



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

TiaBete - Um protótipo de assistente virtual mobile para o autocuidado de pessoas com Diabetes

Trabalho de Conclusão de Curso

Carlos Alberto Santos de Souza

São Cristóvão – Sergipe

2018

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Carlos Alberto Santos de Souza

TiaBete - Um protótipo de assistente virtual mobile para o autocuidado de pessoas com Diabetes

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientadora: Prof^ª Dr^ª Adicinéia Aparecida de Oliveira
Coorientador: Prof. Dr. Lysandro Pinto Borges

São Cristóvão – Sergipe

2018

Carlos Alberto Santos de Souza

TiaBete - Um protótipo de assistente virtual mobile para o autocuidado de pessoas com Diabetes/ Carlos Alberto Santos de Souza. – São Cristóvão – Sergipe, 2018-
109 p. : il.; 30 cm.

Orientadora: Profª Drª Adicinéia Aparecida de Oliveira.

Trabalho de Conclusão de Curso – UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO, 2018.

1. Diabetes *Mellitus*. 2. Assistente Virtual. I. Profª. Drª. Adicinéia Aparecida de Oliveira. II. Universidade Federal de Sergipe. IV. TiaBete - Um protótipo de assistente virtual mobile para o autocuidado de pessoas com Diabetes

CDU

Carlos Alberto Santos de Souza

TiaBete - Um protótipo de assistente virtual mobile para o autocuidado de pessoas com Diabetes

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Trabalho submetido para avaliação. São Cristóvão, 07 de Dezembro de 2018.

Prof^ª. Dr^a. Adicinéia Aparecida de Oliveira
Orientadora

Prof. Dr. Lysandro Pinto Borges
Coorientador

Prof. Ms. Gilton José Ferreira da Silva
Examinador 1

**Prof^ª. Dr^a. Debora Maria Coelho
Nascimento**
Examinador 2

Agradecimentos

Agradeço a Deus primeiramente por ter me dado saúde e força para não desistir dos meus objetivos.

Agradeço a minha família por sempre ter me apoiado mesmo nos momentos mais difíceis.

Agradeço também a Professora Doutora Adicinéia Aparecida Oliveira por ter tanta paciência comigo e por ter me ajudado durante a elaboração deste trabalho com sábios conselhos.

Quero agradecer também ao meu amigo Professor Doutor Lysandro Pinto Borges por me suportar nos meus momentos de extremo stress e pela sua infinita compreensão.

Aos demais amigos e colegas da graduação que tive a oportunidade de conhecer. Foi uma honra trabalhar com todos vocês cavalheiros e damas.

*"Mas os que esperam no Senhor
renovarão as suas forças,
subirão com asas como águias;
correrão, e não se cansarão;
caminharão, e não se fatigarão."
(Isaías 40:31)*

Resumo

O Diabetes foi responsável por cerca de 3,7 milhões de mortes em 2012 e está presente em 425 milhões de pessoas no mundo, sendo que muitas delas não controlam o seu nível de glicose no sangue, o que aumenta o risco de futuras comorbidades do diabetes e assim levar o paciente a óbito. Mas apesar disso é possível que o diabético tenha uma vida longa e de qualidade através do uso de aplicativos móveis projetados para o autocuidado do paciente. Com base nisso, este trabalho tem por objetivo apresentar um protótipo de uma Assistente Virtual em um aplicativo móvel para *smartphones*, chamado de TiaBete, que é voltado para o autocuidado do Diabetes *Mellitus*. Nele foi executada uma revisão sistemática sobre intervenções de aplicativos móveis para o tratamento do Diabetes. Além disso, foi realizada uma pesquisa de campo através de disponibilização de formulários pela internet. Dentre os resultados, descobriu-se que cerca de 90% dos diabéticos são do tipo 2 e que destes apenas 26% controlam o nível de glicose no sangue. Estudos selecionados mostraram que o tratamento com aplicações móveis melhoram os resultados clínicos de pessoas com Diabetes. A pesquisa de campo mostrou que cerca de 87,5% dos entrevistados querem um aplicativo para "*smartphone*" que forneça o controle do Diabetes. Foi elaborada uma pesquisa por aplicações semelhantes, onde foram selecionadas três aplicações da loja de aplicativos da *Google*. Esta pesquisa mostrou que as funcionalidades de tratamento são as mais comuns entre elas, mas nenhuma foca no autocuidado do Diabetes. Desta forma, foi construído um protótipo de aplicativo móvel que demonstra o comportamento de uma assistente virtual criada sobre um modelo de Agente de Software, que monitora informações sobre a aferição da Glicemia Capilar fornecidas pelo paciente.

Palavras-chave: Diabetes *Mellitus*, Assistente virtual, Autocuidado, Complicações, Comorbidades, Aplicativos móveis, Agente de Software.

Abstract

Diabetes accounted for about 3.7 million deaths in 2012 and is present in 425 million people worldwide, and many of them do not control their blood glucose level, which increases the risk of future comorbidities of diabetes and thus lead the patient to death. But the use of diabetics is a long and long form of use of manual applications for patient self-care. Based on this, it is our goal to present a prototype of a virtual assistant, to call the text smartphones, called TiaBete, which is aimed at self-care of Diabetes mellitus. It was a retrospective review of a systematic review of mobile application modalities for the treatment of Diabetes. In addition, a field survey was conducted through the provision of forms over the internet. Among the results found, verify that 90% of diabetics are type 2 and that only these 26 % controls the level of blood glucose. Selected studies were published with the aim of improving the clinical performance of patients with diabetes. A field survey has shown that control is needed to control diabetes. The app has been searched for similar applications, has launched three applications from the *Google* application store. This research has shown that treatment features are more popular among them, but no focus on self-care. to do Diabetes. In this way, a prototype of a software model was created that allows the operation of a virtual assistant created on a software model, which monitors the information on the measurement of Glycemia.

Keywords: Diabetes *Mellitus*, Virtual Assistant, Self Care, Complications, Comorbidities, Mobile Applications, Software Agent.

Lista de ilustrações

Figura 2.1 – Gráfico do número de pessoas com diabetes no mundo.	21
Figura 2.2 – Gráfico de controle glicêmico no Brasil.	24
Figura 2.3 – Gráfico com a presença de mercado dos sistema operacionais (%).	26
Figura 2.4 – Distribuição de telefones móveis no Brasil.	27
Figura 2.5 – Gráfico dos países que mais consumiram aplicativos no mundo em 2017. . .	28
Figura 2.6 – Abordagem de desenvolvimento Nativo.	30
Figura 2.7 – Arquitetura de uma Aplicação Web Móvel.	32
Figura 2.8 – Arquitetura de uma Aplicação Móvel Híbrida.	33
Figura 2.9 – Arquitetura de uma Aplicação Móvel Interpretada.	35
Figura 2.10 – Arquitetura de uma aplicação móvel Multi Compilada.	36
Figura 2.11 – Arquitetura de uma aplicação desenvolvida no Xamarin.	38
Figura 2.12 – Modelo conceitual de um agente.	39
Figura 2.13 – Fases do <i>RUP</i>	42
Figura 2.14 – Notação dos modelos orientados à objetivos sugerido por (CHEAH et al., 2017).	45
Figura 2.15 – Exemplo de uma modelagem com <i>ROADMAP</i>	46
Figura 2.16 – Modelo de i* objetivos do aplicativo <i>Kuching FloraLover</i>	47
Figura 3.1 – Gráfico do percentual de participantes por estado.	49
Figura 3.2 – Gráfico dos participantes por faixa etária.	50
Figura 3.3 – Gráfico do Tipo do diabetes entre os participantes.	51
Figura 3.4 – Gráfico das plataformas móveis dos smartphones dos participantes.	52
Figura 3.5 – Gráfico das funcionalidades mais importantes para os usuários com diabetes.	53
Figura 3.6 – Gráfico dos problemas encontrados em aplicativos.	54
Figura 4.1 – Processo da revisão sistemática.	55
Figura 4.2 – Resultados das etapas de inclusão e exclusão.	58
Figura 4.3 – Artigos selecionados por ano de publicação.	61
Figura 5.1 – Aplicativo mySugr: Diário da Diabetes.	67
Figura 5.2 – Aplicativo mySugr: Diário da Diabetes.	68
Figura 5.3 – Aplicativo Diabetes:M.	69
Figura 5.4 – Demais funcionalidades do Aplicativo Diabetes:M.	69
Figura 5.5 – Aplicativo <i>SocialDiabetes</i>	70
Figura 6.1 – Modelo de Objetivos <i>ROADMAP</i>	73
Figura 6.2 – Modelo de *i Objetivos Nivel Glicemia Capilar.	75

Figura 6.3 – Diagrama de Use Case da aplicação da Tia Bete.	80
Figura 6.4 – Diagrama de <i>Use Case</i> da 1º Iteração.	82
Figura 6.5 – Diagrama de Classes de Domínio da 1º Iteração.	82
Figura 6.6 – Arquitetura de software.	83
Figura 6.7 – Visão de um projeto Android no <i>Visual Studio</i>	85
Figura 6.8 – Diagrama de Classes de Projeto da 1º iteração.	86
Figura 6.9 – Diagrama de Classes de Projeto do Agente TiaBete.	87
Figura 6.10 – Modelo de dados do aplicativo TiaBete.	88
Figura 6.11 – Diagrama de Sequência do Agente Tiabete.	89
Figura 6.12 – Diagrama de Sequência Instanciar <i>TaskFactory</i>	90
Figura 6.13 – Diagrama de Máquina de Estados do Agente TiaBete.	91
Figura 6.14 – Funcionalidade Login da TiaBete.	92
Figura 6.15 – Funcionalidade Cadastro de Paciente.	92
Figura 6.16 – Módulo Glicose.	93
Figura 6.17 – Cadastro de Aferição da Glicemia Capilar.	93
Figura 6.18 – Dicas dadas pelo aplicativo.	94
Figura 6.19 – Animações TiaBete.	95
Figura 6.20 – Mensagens da tarefa EstimacaoHemoglobinaGlicada.	95
Figura 6.21 – Visão do Aplicativo TiaBete dentro do Google Play Console.	96

Lista de quadros

Quadro 2.1 – Atividades metodológicas e tarefas para o desenvolvimento de um SBA. . .	44
Quadro 4.1 – Palavras chaves para construção das strings de busca.	56
Quadro 4.2 – Artigos selecionados.	59
Quadro 4.3 – Técnicas de tratamento do Diabetes aplicadas na intervenção.	60
Quadro 5.1 – Filtros utilizados na Revisão de Mercado.	66
Quadro 5.2 – Descrição dos Aplicativos Aceitos na Revisão de Mercado.	67
Quadro 5.3 – Características extraídas dos aplicativos da Revisão de Mercado.	71
Quadro 6.1 – Requisitos Funcionais do aplicativo.	77
Quadro 6.2 – Requisitos Não-Funcionais do aplicativo.	78
Quadro 6.3 – Principais Atores.	79
Quadro 6.4 – Iterações e seus respectivos casos de uso.	81
Quadro 6.5 – Iterações e seus respectivos casos de uso.	85

Lista de tabelas

Tabela 4.1 – Strings de busca.	57
--	----

Lista de abreviaturas e siglas

ADA	<i>American Diabetes Association</i>
AOT	<i>Ahead-of-Time</i>
API	<i>Application Programming Interface</i>
ART	<i>Android Runtime</i>
AVC	Acidente Vascular Cerebral
DAC	Doença Arterial Coronariana
DAP	Doença Arterial Periférica
DCV	Doença Cardiovascular
DM	Diabetes <i>Mellitus</i>
DM1	Diabetes <i>Mellitus</i> tipo 1
DM2	Diabetes <i>Mellitus</i> tipo 2
DMG	Diabetes <i>Mellitus</i> Gestacional
GPS	Sistema de Posicionamento Global
HbA1c	Hemoglobina Glicada
HAL	<i>Hardware Abstraction Layer</i>
IA	Inteligência Artificial
IDE	<i>Integrated Development Environment</i>
IDF	<i>International Diabetes Federation</i>
JIT	<i>Just-in-Time</i>
MVP	<i>Minimum Viable Product</i>
OHA	<i>Open Handset Alliance</i>
PU	<i>Process Unified</i>
RF	Requisito Funcional
RNF	Requisito Não - Funcional

RS	Requisitos de Sistema
RU	Requisitos de Usuário
RUP	<i>Rational Unified Process</i>
SBAS	Sistema Baseado em Agentes de Software
SBD	Sociedade Brasileira de Diabetes
SCA	Sistema Centrado em Agentes
SO	Sistema Operacional
TTG	Teste de Tolerância à Glicose
WHO	<i>World Health Organization</i>
UI	<i>User Interface</i>

Sumário

1	Introdução	17
1.1	Objetivos	18
1.1.1	Objetivo Geral	18
1.1.2	Objetivos Específicos	18
1.2	Metodologia	19
1.3	Estrutura do Documento	19
2	Fundamentação teórica	20
2.1	Diabetes <i>Mellitus</i>	20
2.1.1	Comorbidades e Complicações	23
2.2	Autocuidado	24
2.3	Aplicações Móveis	25
2.3.1	Aplicações Móveis Nativas	29
2.3.2	Aplicações Móveis Multiplataforma	31
2.3.3	Aplicações Web Móveis	32
2.3.4	Aplicações Móveis Híbridas	33
2.3.5	Aplicações Móveis Interpretadas	34
2.3.6	Aplicações Móveis Multi Compiladas	35
2.4	Android	37
2.5	Xamarin	38
2.6	Agentes de Softwares	39
2.6.1	Tipos de Agentes de Software	40
2.7	Desenvolvimento de Software	41
2.7.1	RUP	42
2.7.2	Processo de desenvolvimento de Agentes de Software para dispositivos móveis	43
2.7.3	ROADMAP e modelos de i* Objetivos	45
2.8	Considerações do capítulo	47
3	Pesquisa de campo	49
3.1	Resultados da pesquisa	49
3.2	Considerações do capítulo	54
4	Revisão sistemática	55
4.1	Metodologia da Revisão Sistemática	56
4.2	Resumos dos artigos selecionados	61

4.2.1	<i>Developing a behavioral model for mobile phone-based diabetes interventions</i>	61
4.2.2	<i>A lifestyle intervention supported by mobile health technologies to improve the cardiometabolic risk profile of individuals at risk for cardiovascular disease and type 2 diabetes: study rationale and protocol . . .</i>	62
4.2.3	<i>A mobile diabetes management and educational system for type- 2 diabetics in Saudi Arabia (SAED)</i>	63
4.2.4	<i>A randomized wait-list control trial to evaluate the impact of a mobile application to improve self-management of individuals with diabetes: a study protocol</i>	63
4.2.5	<i>I don't have to explain, people understand": acceptability and cultural relevance of a mobile health lifestyle intervention for filipinos with type 2 diabetes</i>	64
4.2.6	<i>Users' thoughts and opinions about a self- regulation-based eHealth intervention targeting physical activity and the intake of fruit and vegetables: A qualitative study</i>	65
4.3	Considerações do capítulo	65
5	Estudo de Aplicações Semelhantes	66
5.1	Metodologia de Pesquisa	66
5.2	mySugr: Diário da Diabetes	67
5.3	Diabetes:M	68
5.4	SocialDiabetes	70
5.5	Considerações sobre os aplicativos semelhantes	70
5.6	Considerações do capítulo	71
6	Desenvolvimento do aplicativo	72
6.1	Visão Geral do Produto e Projeto	72
6.1.1	Requisitos Funcionais	76
6.1.2	Requisitos Não-Funcionais	78
6.1.3	Modelagem de Negócio	79
6.2	Fase de Concepção da 1º Iteração	81
6.3	Fase de Elaboração da 1º Iteração	83
6.4	Fase de Construção da 1º Iteração	84
6.4.1	Ferramentas e Especificações Técnicas	84
6.4.2	Construção do Código	86
6.4.3	Demonstração	91
6.5	Transição	96
6.6	Considerações do capítulo	96

7	Considerações Finais	98
	Referências	101
	Apêndices	105
	Apêndice A - Questionário usado na pesquisa de campo para elicitaco de requisitos do aplicativo.	106

1 Introdução

O Diabetes *Mellitus* (DM) é uma doença grave e crônica que merece atenção em todo o mundo, sendo que 90% dos casos de DM são do tipo 2 (OLIVEIRA, 2017). Desde o seu primeiro caso constatado no Egito à 1500 a. C., a quantidade de casos da doença têm aumentado ao longo dos anos (GAMA, 2002 apud LISANGELA, 2013).

Um relatório publicado pela International Diabetes Federation (IDF), reportou que cerca de 425 milhões de adultos no mundo possuíam Diabetes em 2017 (INTERNATIONAL DIABETES FEDERATION, 2017). Dentre estas pessoas com Diabetes, 327 milhões estão na faixa dos 20 à 64 anos, enquanto que na faixa dos 65 à 79 anos somam 98 milhões de pessoas no mundo. O relatório mostra ainda que, se as tendências atuais persistirem, segundo projeções, esse número tende a aumentar para 629 milhões em 2045 (INTERNATIONAL DIABETES FEDERATION, 2017). Segundo dados da *World Health Organization* (WHO), em 2012, o Diabetes causou 3,7 milhões de mortes. Dentre estas mortes, 59% foram ocasionadas devido ao alto nível de glicose no sangue, fator que aumentou os riscos de doenças cardiovasculares e outras doenças, sendo que 43% das mortes ocorreram em pessoas com menos de 70 anos de idade (WHO, 2016).

Apesar do Diabetes ser uma doença crônica, é possível que a pessoa com diabetes tenha uma vida longa e de qualidade. Esta meta pode ser alcançada por meio de monitoramento e controle da glicemia, alimentação saudável, práticas regulares de exercícios físicos e uso de medicação adequada (SOCIEDADE BRASILEIRA DE DIABETES, 2018). A falta de atenção a estas variáveis pode provocar o surgimento de várias comorbidades que provém de complicações como hipoglicemia, cetoacidose diabética, retinopatia, nefropatia, doença arterial coronariana (DAC), infarto do miocárdio, doença arterial periférica (DAP), acidente vascular cerebral (AVC), encefalopatia diabética e pé diabético (LISANGELA, 2013).

Segundo a Associação Americana de Diabetes (ADA) as técnicas de tratamento convencionais podem criar barreiras para o paciente de baixa renda e com pouco conhecimento no tratamento da Diabetes, como: depressão, pobreza, barreiras relacionadas às responsabilidades e dinâmicas familiares (AMERICAN DIABETES ASSOCIATION, 2015). Cuidar de doenças crônicas como o Diabetes requer um modelo diferente, no qual os pacientes podem ser fortalecidos por meio do autocuidado da Diabetes (HSU et al., 2016). Isto pode ser feito por intermédio de uso de aplicativos móveis que mantêm registros sobre a alimentação, histórico dos níveis de glicose no sangue, interação medicamentosa, atividade física e o modo como o paciente está se sentindo (SOCIEDADE BRASILEIRA DE DIABETES, 2018).

Este trabalho apresenta o processo de pesquisa e desenvolvimento de uma assistente virtual para dispositivos móveis voltada para o autocuidado do Diabetes chamada TiaBete.

Para isto, primeiramente foi feito um embasamento teórico sobre todos os termos e conceitos envolvidos no projeto. Depois, foi realizada uma pesquisa de campo por meio de formulários web pela internet, para eliciação dos requisitos de software. Em seguida, foi produzida a revisão sistemática sobre casos de estudo envolvendo o uso de aplicações móveis. Após terminada a revisão sistemática, foi produzido um estudo de aplicações semelhantes que analisou aplicativos e comparou-as em relação às suas funcionalidades. Por fim, foi construído um protótipo de uma assistente virtual para aplicativos móveis, desenvolvido para a plataforma *Android*.

1.1 Objetivos

Nesta seção são apresentados os objetivos gerais e específicos para realização do corrente trabalho de conclusão de curso.

1.1.1 Objetivo Geral

O Objetivo Geral deste trabalho foi construir um protótipo de uma Assistente Virtual para o autocuidado de pacientes com Diabetes, implementado como um aplicativo móvel com base em um agente de software.

1.1.2 Objetivos Específicos

Para se atingir o objetivo proposto, foram delineados os seguintes objetivos específicos:

- Construir um comparativo entre as abordagens de desenvolvimento de aplicações móveis;
- Prover uma análise de requisitos funcionais e não - funcionais, provenientes de dados obtidos na pesquisa de campo;
- Fornecer uma revisão sistemática para obter o estado da arte sobre tecnologias de aplicações móveis para o autocuidado de pessoas com DM;
- Levantar um comparativo entre funcionalidades de aplicações semelhantes presentes na Loja Virtual do *Google Play*;
- Proporcionar o protótipo de uma Assistente Virtual para dispositivos móveis.

1.2 Metodologia

Este trabalho possui caráter quantitativo e exploratório. Ele é considerado quantitativo por prover dados da pesquisa de campo realizada por meio de formulários digitais, publicados pela internet. Este trabalho é considerado exploratório, pois este trabalho visa criar uma nova perspectiva de autocuidado da pessoa com diabetes. A aplicação de software está em conformidade com as diretrizes propostas pela Sociedade Brasileira de Diabetes (OLIVEIRA, 2017) e recomendações da Secretaria de Atenção à Saúde, órgão ligado ao Ministério da Saúde do Brasil (BRANDÃO; PINHEIRO, 2016). O aplicativo foi desenvolvido para *smartphones Android* através do paradigma multicompileado, com o uso do modelo de processo incremental e interativo RUP (SOMMERVILLE, 2011) aliado ao desenvolvimento de Aplicações móveis Baseados em Agentes de Software, orientado a Casos de Uso, que foi auxiliado com artefatos UML (PRESSMAN; MAXIM, 2016).

1.3 Estrutura do Documento

O texto deste trabalho será organizado em 6 capítulos além desta introdução. Os tópicos a seguir descrevem o conteúdo de cada um dos deles, que são:

- Capítulo 2 - Fundamentação Teórica: explana todo o referencial teórico necessário para o entendimento e elaboração do trabalho. Nele são abordados os conceitos sobre o Diabetes e Autocuidado, bases do desenvolvimento de Aplicações móveis e o uso de agentes de software;
- Capítulo 3 - Pesquisa de Campo: descreve os resultados obtidos por meio de consulta pública, com uso de formulários disponibilizados na internet para pacientes com diabetes;
- Capítulo 4 - Revisão Sistemática: discorre sobre o processo metodológico, resultados e análise dos artigos selecionados para demonstrar o estado da arte das tecnologias móveis para o autocuidado do DM;
- Capítulo 5 - Estudo de Aplicações Semelhantes: relata o passo a passo da pesquisa por aplicações semelhantes feita na loja de aplicativos da *Google Play*;
- Capítulo 6 - Desenvolvimento do aplicativo: descreve o processo de construção do aplicativo tema deste trabalho;
- Capítulo 7 - Considerações Finais: resume todos os achados do trabalho, descreve os resultados obtidos e trabalhos futuros.

2 Fundamentação teórica

O Diabetes *Mellitus* é uma doença que pode levar a morte e não tem cura (LISANGELA, 2013). No mundo cerca de 425 milhões de pessoas possuem algum tipo de diabetes, sendo que cerca de 3,7 milhões de pessoas já morreram por causa do Diabetes em 2012 (WHO, 2016). A maior parte das pessoas com Diabetes não controlam corretamente o seu nível de glicose no sangue, isto ocorre por falta de conhecimento e recursos para utilização das técnicas de autocuidado (NETTO, 2015). Como forma de aumentar o conhecimento da pessoa com Diabetes sobre o mesmo, este trabalho construiu um protótipo de uma assistente virtual focado no autocuidado do Diabetes. Neste capítulo abordam-se estudos que fundamentam os principais conceitos relacionados ao projeto.

Inicia-se definindo o Diabetes *Mellitus*, detalhando o Diabetes *Mellitus* quanto o nível de abrangência dentre as pessoas com diabetes e suas características peculiares. Logo após, apresentam-se as principais comorbidades e complicações, e prossegue-se descrevendo os conceitos relacionados ao autocuidado do paciente com diabetes.

Em continuidade com o capítulo, conceitua-se as aplicações móveis, descrevendo as características das aplicações móveis nativas, aplicações móveis multiplataforma, *web* móveis, aplicações híbridas, aplicações interpretadas e por fim as aplicações Multi Compiladas. Em continuidade, são abordadas as principais ferramentas de desenvolvimento da plataforma móvel *Android*, a exemplo, *Xamarin*.

Os conceitos de Agentes de Software e processo de Engenharia de Software para Sistemas Baseados em Agentes para aplicações móveis são descritas logo em seguida. No último tópico é descrita a metodologia de processo de desenvolvimento utilizado neste projeto. Para apoiar os conceitos de construção de Agentes para Aplicações móveis, o *RUP* foi o modelo de processo utilizado em sua construção.

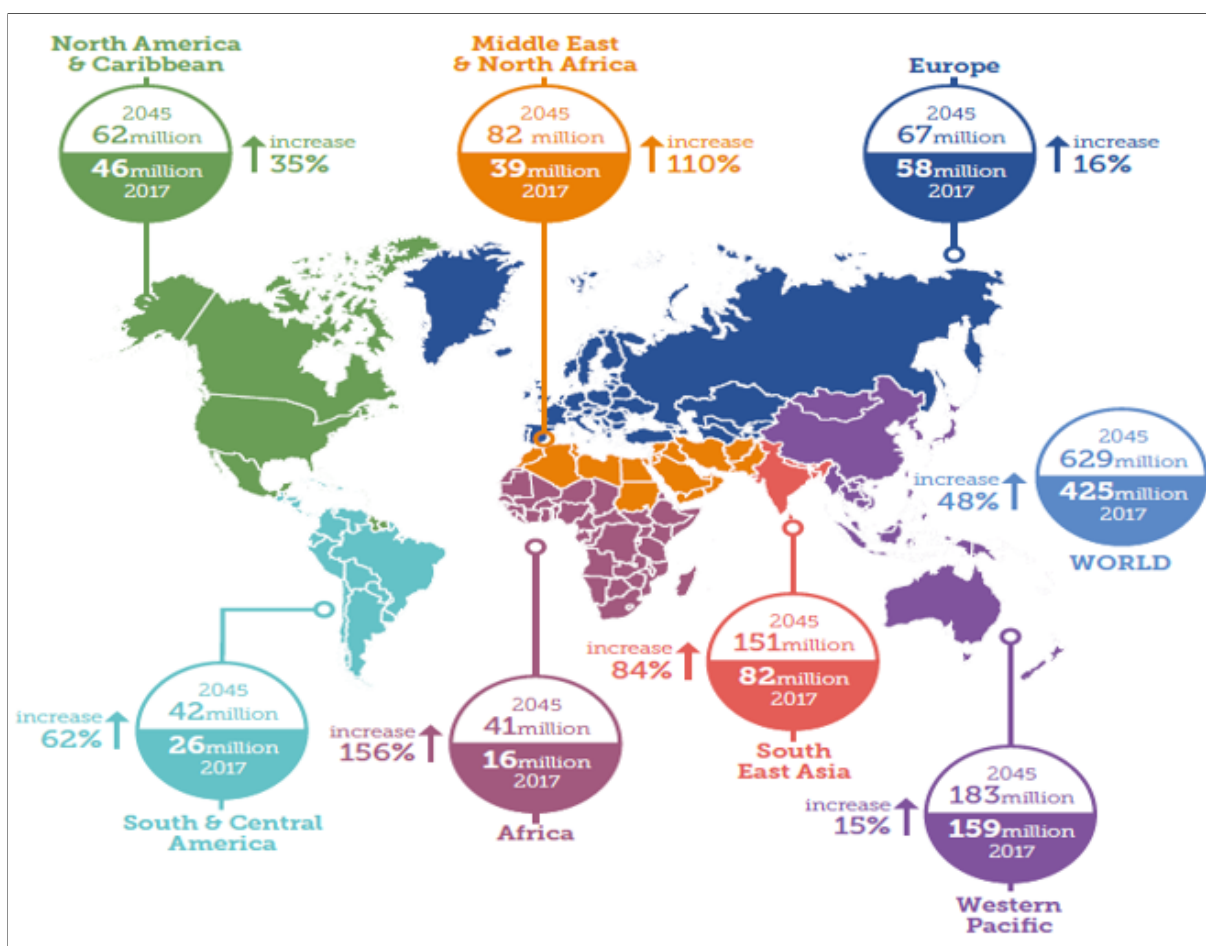
2.1 Diabetes *Mellitus*

Segundo (GAMA, 2002 apud LISANGELA, 2013), o primeiro caso constatado do Diabetes ocorreu no Egito em 1500 a.C., mas por sua vez era ainda tida como uma doença desconhecida. O termo Diabetes foi citado pela primeira vez por Apolônio e *Memphis* em 250 a.C., que significa sifão em grego, que é o mesmo que dizer tubo para aspirar a água, devido ao sintoma que provoca sede intensa e elevada quantidade de urina. O Diabetes recebe a denominação *Mellitus* no século 1 d.C., que do latim, significa mel, que depois é conhecida como urina doce.

O Diabetes *Mellitus* (DM) é definido pela Sociedade Brasileira de Diabetes (SBD) como sendo uma doença que afeta a produção ou uso da insulina pelo corpo. Neste caso, o corpo não produz insulina ou não consegue utilizar efetivamente a insulina que produz. A insulina é o hormônio responsável pelo controle da quantidade de glicose no sangue. Através dela o corpo transforma a glicose proveniente dos alimentos ingeridos em energia, que por sua vez é essencial para a vida humana (SOCIEDADE BRASILEIRA DE DIABETES, 2018).

A Figura 2.1, exibe um gráfico extraído do atlas de DM, publicado pela *International Diabetes Federation* (IDF). Ele exibe a quantidade de pessoas com diabetes no mundo e suas projeções para 2045.

Figura 2.1 – Gráfico do número de pessoas com diabetes no mundo.



Fonte: *International Diabetes Federation* (2017).

Como mostra a Figura 2.1, no mundo existem cerca de 425 milhões de pessoas diagnosticadas com diabetes em 2017 e este valor tende a aumentar, segundo projeções, para 629 milhões em 2045. Segundo a *World Health Organization* (WHO), o DM causou 1 milhão e meio de mortes em 2012. Cerca de 2,2 milhões de mortes foram ocasionadas devido ao descontrole do nível de glicose no sangue. Cerca de 43% destes 3,7 milhões de mortes ocorreram antes dos 70

anos de idade. Ela complementa ainda que as mortes de pessoas antes dos 70 anos ocorrem mais em países de baixa renda (WHO, 2016).

Conforme Brandão e Pinheiro (2016), o diagnóstico do diabetes é feito através da detecção de hiperglicemia, ou seja, níveis elevados de glicose no sangue. Este diagnóstico é feito através de testes como: glicemia casual, glicemia de jejum, teste de tolerância à glicose com sobrecarga de 75 g em duas horas (TTG) e, em alguns casos, hemoglobina glicada (HbA1c).

Para a *American Diabetes Association* 2015, o DM é classificado de várias formas, as principais são:

- **Diabetes Mellitus tipo 1 (DM1):** neste tipo de diabetes o sistema imunológico do corpo ataca células beta do pâncreas, desta forma, o corpo produz pouca ou nenhuma insulina com relativa ou absoluta deficiência;
- **Diabetes Mellitus (DM2):** é quando o corpo não produz insulina o suficiente ou não consegue utilizar adequadamente a insulina que produz, como consequência, ocorre o aumento da concentração de glicose no sangue, o que prejudica a circulação;
- **Diabetes Mellitus gestacional (DMG):** o corpo da mulher tende a compensar o aumento da produção de hormônios que ocorre durante a gravidez produzindo mais insulina. Em certas mulheres, a compensação não ocorre aumentando assim os níveis de glicose no corpo.

Existe ainda o termo Pré-diabetes, que é na verdade um estágio que antecede o surgimento da DM2 (SOCIEDADE BRASILEIRA DE DIABETES, 2018). Uma pessoa é diagnosticada como pré-diabética se esta apresenta níveis altos de glicose no sangue mas não o bastante para ser diagnosticada como DM2.

A WHO descreve a DM tipo 2 como uma vertente da diabetes onde o corpo humano não utiliza efetivamente a insulina que produz. Ela complementa ainda que os sintomas podem se assemelhar aos sintomas do DM tipo 1 como poliúria, polidipsia, perda inexplicada de peso e polifagia, mas eles são menos aparentes ou até mesmo ausentes na maioria dos casos. Por isso a doença pode não ser diagnosticada por muitos anos, até que as complicações já tenham surgido (WHO, 2016).

Segundo Oliveira (2017), cerca de 90% dos pacientes com DM no mundo são do tipo 2. Esta doença acomete pacientes geralmente a partir dos 40 anos de idade, mas existem casos de pessoas mais jovens que foram diagnosticadas com a doença. Em 80% a 90% dos casos, associa-se ao excesso de peso e a outros componentes da síndrome metabólica.

Brandão e Pinheiro (2016) apontam que o tratamento da diabetes tipo 2 consiste na adoção de hábitos saudáveis, dieta balanceada, prática regular de exercícios físicos, moderação

no uso de álcool, abandono do tabagismo e tratamento farmacológico. Todos estes hábitos corroboram para o controle glicêmico que é o objetivo principal a ser atingido pelo diabético.

2.1.1 Comorbidades e Complicações

Quando o nível glicêmico permanece alto durante muito tempo da vida do paciente, complicações de saúde tendem a surgir e assim provocar o surgimento de outras doenças e comorbidades (LISANGELA, 2013).

Segundo (INTERNATIONAL DIABETES FEDERATION, 2017) principais complicações do DM são:

- **Aumento do risco de Doença Cardiovascular (DCV):** altos níveis de glicose no sangue tornam o sistema de coagulação sanguínea mais ativa, o que aumenta o risco de surgir coágulos. A DM também está associada a altos níveis de pressão arterial e colesterol;
- **Perda de visão (retinopatia):** presença de grandes quantidades de glicose no sangue prejudicam os vasos capilares presentes no olhos, levando a hemorragia e ao bloqueio capilar, que como consequência, leva a perda de visão;
- **Problemas renais (nefropatia):** doenças nos rins são muito comuns em pessoas com diabetes com níveis de glicose altos, ela pode levar a incidências de infecções no trato urinário;
- **Problemas nos nervos (neuropatia):** altas taxas de glicose no sangue podem causar problemas nos nervos em todo o corpo. Esta complicação pode se passar despercebida, causando doenças como disfunção autonômica cardíaca e o aparecimento de úlceras;
- **Pé diabético:** é complicação crônica e grave que consiste no aparecimento de lesões em tecidos profundos dos pés associadas com problemas neurológicos. Em estágio muito avançado, leva a amputação dos membros inferiores.

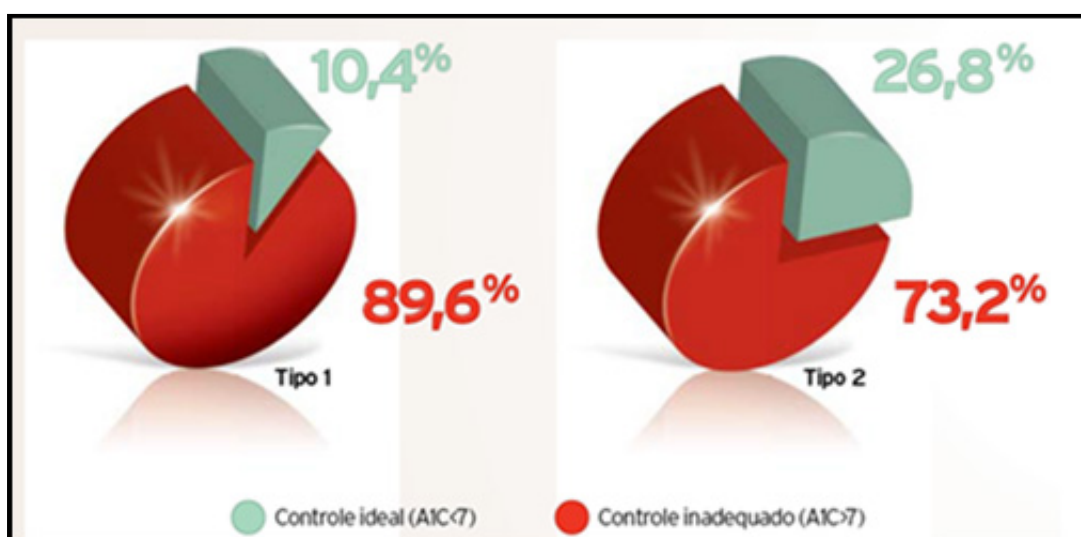
Conforme Oliveira (2017), estas complicações se não evitadas ou tratadas, podem levar ao aparecimento de comorbidades como a hipertensão, perda de visão, complicações do coração (surgimento de dislipidemias que são relacionadas ao colesterol), neuropatia, acidente vascular cerebral (AVC), como também a amputação de membros inferiores (fruto do agravamento das lesões nos mesmos).

2.2 Autocuidado

Apesar da DM ser uma doença crônica e grave, é possível ter uma vida com qualidade. Para WHO (2016), um bom gerenciamento dos índices glicêmicos podem prevenir complicações e a morte prematura, isto é possível graças ao uso de intervenções que promovem estilos de vida mais saudáveis, como rastreamento e tratamento regular, para detecção precoce de complicações, que são frutos de uma boa educação do paciente quanto ao autocuidado.

A Figura 2.2 mostra o resultado de um estudo feito no Brasil com 6600 pacientes com DM tipo 1 e tipo 2, distribuídos em 12 centros de diabetes localizados em diferentes regiões do país.

Figura 2.2 – Gráfico de controle glicêmico no Brasil.



Fonte: Netto (2015).

Como relatado na Figura 2.2, respectivamente, cerca de 89,6% e 73,2% dos pacientes com DM tipo 1 e tipo 2 não controlam o nível de glicemia de forma adequada. Isto é um grave problema, pois, pode ocasionar o aparecimento de complicações da diabetes e assim estes pacientes correm maior risco de morte do que os que possuem a glicemia controlada.

Segundo Jongh (2012), o autocuidado é um conjunto de tarefas que uma pessoa pode executar para minimizar o impacto negativo que uma doença pode causar a sua saúde. Ele complementa ainda que o autocuidado requer que a pessoa tenha habilidades de automonitoramento dos sintomas, conhecimento das implicações, para assim ajustar a sua medicação, tratamento e comportamentos.

Abaixo são abordadas as principais tecnologias usadas para o autocuidado do DM (OLIVEIRA, 2017):

- **Bombas de infusão de insulina:** consiste em um aparelho com comando eletrônico semelhante a um *pager* que possui um reservatório de insulina e um cateter pelo qual ocorre a administração de insulina pelo tecido subcutâneo do paciente. As bombas injetam insulina no corpo do paciente de forma programada;
- **Sistemas de monitoramento contínuo de glicose (CGMS):** é um sistema que registra os níveis de glicose no tecido celular subcutâneo por meio de uma pequena cânula inserida sob a pele. Ela é capaz de medir também o fluído intersticial que correlaciona bem com a glicose plasmática;
- **Sensores de uso pessoal e de medida em tempo real:** nestes equipamentos são exibidos gráficos do padrão glicêmico que indicam tendências de elevação ou queda do nível de glicose no sangue. Eles podem ainda emitir alarmes sonoros em caso de hipo e hiperglicemia;
- **Aplicativos para *smartphones*:** são programas geralmente usados para registrar os níveis de glicose no sangue, calcular as doses de insulina, gerenciar hábitos saudáveis entre outras funcionalidades. São executados em aparelhos móveis, mais comuns em *smartphones*.

Estudos têm mostrado melhorias em resultados clínicos de pacientes que utilizam aplicativos móveis para melhorar o autocuidado da DM (NUNDY et al., 2013; ALOTAIBI; IS-TEPANIAN; PHILIP, 2016). A próxima seção trará informações sobre os aplicativos móveis, detalhando sobre as suas características e justificando a sua popularidade em torno de pacientes com DM, que pode servir como uma ferramenta para baixar os custos com tratamento do Diabetes *Mellitus* Tipo 2.

2.3 Aplicações Móveis

Cada vez mais pessoas no mundo estão usando telefones celulares, seja para uso pessoal ou para desempenhar tarefas do cotidiano (LECHETA, 2013). Esta consequência surgiu a partir da constante evolução de *hardware* e *software* destes aparelhos. Eles estão apresentando cada vez mais recursos tecnológicos sofisticados como câmeras, músicas, *bluetooth*, ótimas interfaces visuais, jogos, uso do *Global Positioning System* (GPS), acesso à internet, *emails* e uso da TV digital.

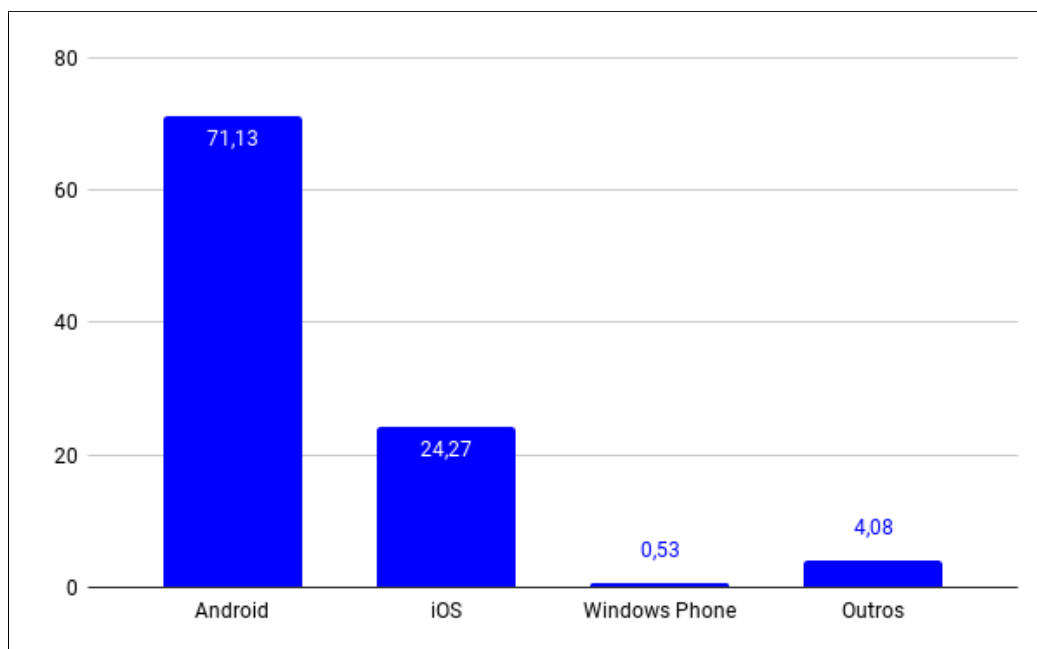
Simões e Pereira (2014) declaram que, além da criação de novos recursos, os celulares móveis evoluíram seus *hardwares*, aumentando a sua capacidade de processamento. Isto permitiu

a execução de aplicações de *software* que antes eram somente comuns aos computadores pessoais. Essa nova geração de aparelhos celulares, batizados de *smartphones*, fornecem maior comodidade aos seus usuários através da disponibilização de aplicações móveis. Neste estudo eles serão também chamados de dispositivos móveis.

Segundo Pressman e Maxim (2016), aplicações móveis ou *mobile* são programas de *softwares* projetados especificamente para os dispositivos móveis com sistemas operacionais como *Android*, *iOS* ou *Windows Phone*, que são também chamados de plataformas móveis. Elas se utilizam de mecanismos de interação exclusivos da plataforma móvel, como também são consumidoras de serviços web através de *browsers* adaptados ao aparelho, interagem com diversos tipos de *hardwares* presentes no dispositivo (acelerômetro ou localização por GPS) e armazenam dados de forma persistente em seu armazenamento local.

Um relatório produzido pela OHA (2018) mostrou que cerca de 3 bilhões de pessoas no mundo possuem um telefone celular. Segundo Jobe (2013), para poder conquistar esse mercado consumidor, as empresas estão cada vez mais interessadas no desenvolvimento de aplicações móveis. Ainda diz que, visando atender a esta demanda, grandes empresas de tecnologia criaram suas próprias plataformas móveis, como por exemplo, o *Android*¹ da *Google*, o *iOS*² da *Apple* e *Windows Phone*³ da *Microsoft*, que são também as plataformas mais populares no mercado mundial. A Figura 2.3 mostra a porcentagem de aparelhos com sistema operacional Android, iOS e Windows Phone no mundo.

Figura 2.3 – Gráfico com a presença de mercado dos sistema operacionais (%).



Fonte: Statcounter (2018).

¹ <https://www.android.com/>

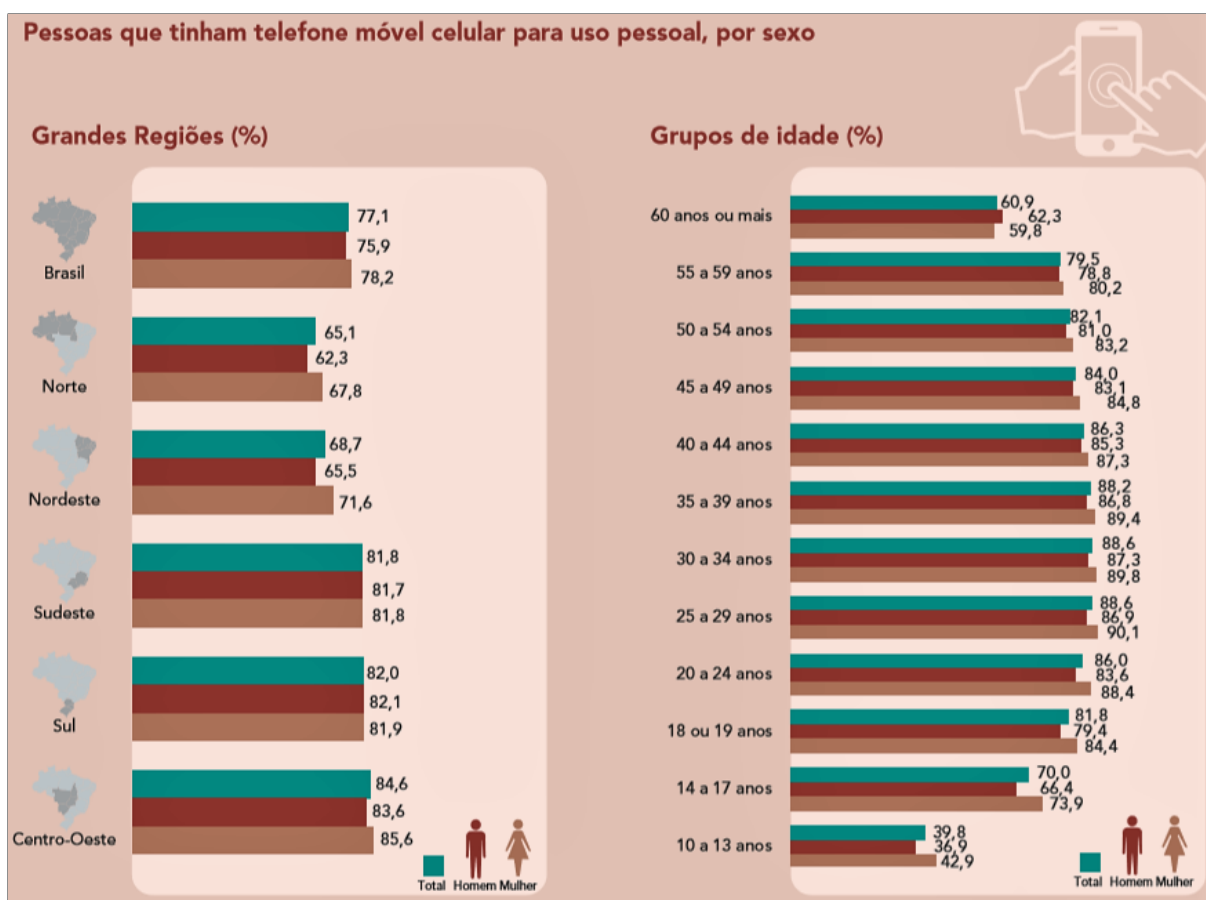
² <https://www.apple.com/br/ios/ios-11/>

³ <https://www.microsoft.com/en-us/store/b/mobile?icid=CNavDevicesMobile>

A Figura 2.3 indica que o *Android* é a plataforma de maior popularidade. Segundo esse relatório, o *Android* está presente em 71.13% dos dispositivos móveis do mundo. Atrás dele está o *iOS* com aproximadamente 24.27%, enquanto que o *Windows Phone* possui 0,53% do mercado *mobile*.

A Figura 2.4 demonstra os resultados da pesquisa feita pelo IBGE sobre o uso de telefones móveis no Brasil, em 2016.

Figura 2.4 – Distribuição de telefones móveis no Brasil.



Fonte: IBGE (2016).

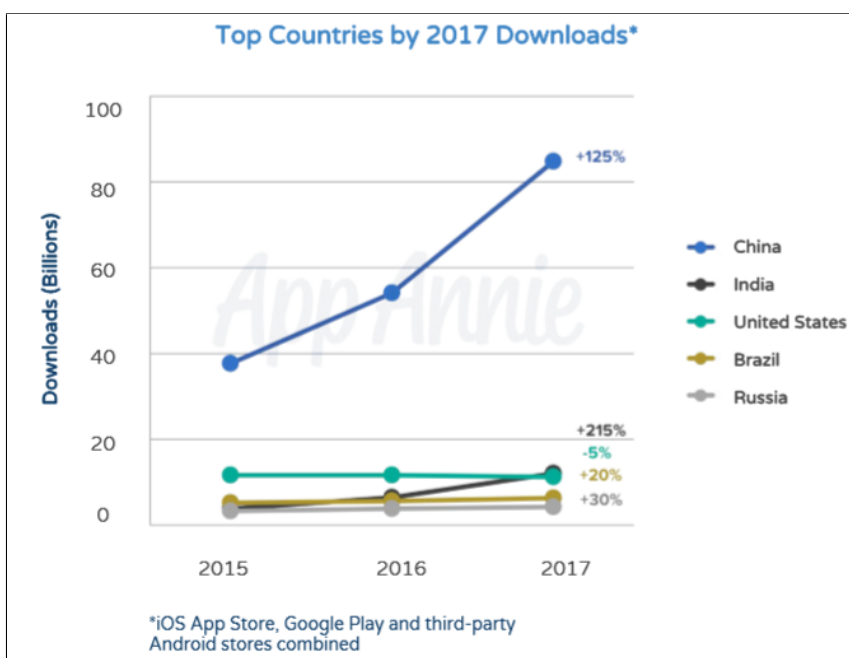
Como mostra a Figura 2.4, cerca de 77,1% da população brasileira tinha telefone móvel para uso pessoal. Na região Nordeste, 65,1% dos entrevistados possuíam aparelho celular, enquanto que nas outras regiões o percentual ultrapassou os 80%. Quanto ao uso de aparelhos celulares por faixa etária, o menor resultado apareceu na faixa de 10 a 13 anos com 39,8%, a maior representatividade ocorreu entre os adultos de 25 à 29 e 30 à 34 anos, que apresentaram a mesma porcentagem de 88,6%. As pessoas entre 35 à 39 anos totalizaram 88,2%, e aproximadamente 60,9% entre idosos de 60 anos ou mais de idade.

Segundo relatório feito pela *App Annie* (2018), cerca de 90% dos *smartphones* do Brasil possuem Sistema Operacional (SO) *Android*, enquanto que nos 10% restantes estão os demais

SOs como *iOS* e *Windows Phone*. Isto acontece por que os aparelhos com *Android* são mais baratos que os *smartphones* com *iOS* e fornecem maior suporte e funcionalidades que os do *Windows Phone* (AVAZU, 2016).

Acompanhando a evolução dos *smartphones*, o mercado mundial de aplicativos móveis também cresceu ao longo dos anos, chegando a ultrapassar a marca de 175 bilhões de *downloads* em 2017 (APPANNIE, 2018). Este mercado movimentou cerca de \$86 bilhões de dólares em 2017 em todo o mundo. A Figura 2.5 demonstra um comparativo feito pela AppAnnie (2018), onde são comparados os consumos de aplicativos móveis dos países que mais baixaram aplicativos móveis no mundo entre os anos 2015, 2016 e 2017.

Figura 2.5 – Gráfico dos países que mais consumiram aplicativos no mundo em 2017.



Fonte: AppAnnie (2018).

Na Figura 2.5 o Brasil encontra-se em quarto lugar, abaixo da China, Índia e Estados Unidos. Estes indicadores foram retirados das lojas de aplicativos *Apple Store*⁴ para *iOS* e *Google Play*⁵ para *Android*.

Dentre os países que baixam mais aplicativos pela *App Store*, o Brasil se encontra em nono lugar, atrás da China, Estados Unidos, Japão, Inglaterra, Rússia, França, Alemanha e Canadá. Já no ranking dos países que mais consumiram aplicativos pela *Google Play*, o Brasil aparece em segundo, perdendo apenas para a Índia. Isto acontece justamente porque a maioria dos aparelhos vendidos no Brasil possuem SO *Android*. (APPANNIE, 2018).

⁴ <https://www.apple.com/br/ios/app-store/>

⁵ https://play.google.com/store?hl=pt_BR

Fernandes e Ferreira (2016) sugerem que, para assegurar o crescimento deste mercado, é necessário que sejam desenvolvidos aplicativos de alta qualidade que supram as expectativas e necessidades dos consumidores. O brasileiro possui uma média de 63 aplicativos instalados em seus dispositivos, mas apenas 22 são realmente utilizados (AVAZU, 2016).

Existem dois principais paradigmas de desenvolvimento móvel, o Nativo e o Multiplataforma (QUE; GUO; ZHU, 2017; LATIF et al., 2017; JOBE, 2013). Cada paradigma apresenta suas próprias abordagens de desenvolvimento. O paradigma Nativo traz a abordagem de desenvolvimento de aplicações nativas, ou seja, aplicações que são projetadas para especificamente uma plataforma móvel. Já as abordagens de aplicações Web Móveis, Híbridas, Interpretadas e Multi Compiladas são construídas a partir do paradigma Multiplataforma, onde geram aplicativos para várias plataformas de uma só vez a partir de um único código, (LATIF et al., 2017; BIØRN-HANSEN; MAJCHRZAK; GRØNLI, 2017). Cada paradigma possui um conjunto de pontos fortes e fracos que são comuns as abordagens que elas abrangem. Estas características devem ser levadas em consideração antes de se iniciar a construção de um aplicativo móvel (QUE; GUO; ZHU, 2017; BIØRN-HANSEN; MAJCHRZAK; GRØNLI, 2017; JOBE, 2013).

2.3.1 Aplicações Móveis Nativas

Na abordagem de desenvolvimento de aplicações nativas, os desenvolvedores utilizam diversas ferramentas, linguagens e padrões próprios da plataforma, de modo que o resultado é um aplicativo móvel compatível apenas a uma plataforma móvel (QUE; GUO; ZHU, 2017). Estas aplicações estão sendo desenvolvidas por meio de pelo menos três linguagens de programação: Java para o Android, Objective-C para o iOS e C# para Windows Phone (PRESSMAN; MAXIM, 2016). As plataformas de desenvolvimento mais comuns para desenvolvimento nativo são: *Eclipse*⁶ ou Android Studio⁷ para o Android, Xcode⁸ para o iOS e Visual Studio⁹ para o Windows Phone (LATIF et al., 2016; LATIF et al., 2017).

Jobe (2013) afirma que as aplicações nativas são melhores na criação de conteúdo e apresentam melhor performance em relação ao *hardware* do dispositivo. As características comuns entre elas é que todas têm acesso direto ao *hardware* do dispositivo, dá suporte em todas as interfaces do usuário e interagem com todo o ambiente da plataforma móvel. Isto facilita a construção de aplicações que focam na criação de conteúdo e que utilizam vários componentes de *hardware* ao mesmo tempo.

⁶ <http://www.eclipse.org/downloads/eclipse-packages/>

⁷ <https://developer.android.com/studio/index.html?hl=pt-br>

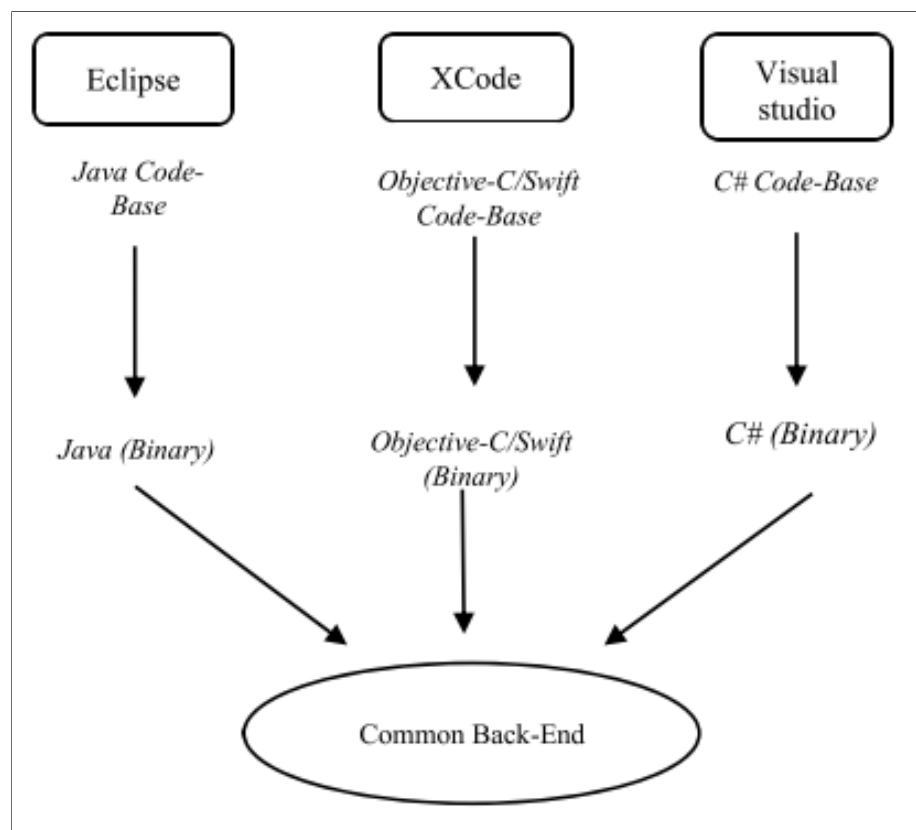
⁸ <https://developer.apple.com/xcode/>

⁹ <https://www.visualstudio.com/pt-br/?rr=https%3A%2F%2Fwww.google.com.br%2F>

Que, Guo e Zhu (2017) descrevem que na programação nativa, a reutilização de código para outras plataformas é fraca, visto que as ferramentas e linguagens que são utilizadas no desenvolvimento das aplicações são suportadas somente em uma plataforma específica, as aplicações devem ser desenvolvidas separadamente para cada plataforma móvel. Desta forma, os projetos acabam demandando muito mais tempo e esforço que nas demais abordagens. Além disso, o desenvolvimento se torna mais complexo por necessitar de conhecimento de vários sistemas operacionais.

A Figura 2.6 exibe uma abstração de como o código fonte de uma aplicação móvel se comporta durante o seu desenvolvimento até a sua implantação.

Figura 2.6 – Abordagem de desenvolvimento Nativo.



Fonte: Latif et al. (2016).

A Figura 2.6 mostra na parte superior alguns dos principais ambientes de desenvolvimento nativo de cada plataforma móvel (*Eclipse*, *XCode* e *Visual studio*). Estes ambientes de desenvolvimento utilizam código específico em suas respectivas plataformas móveis, como é o caso do *Java*, *Objective-C/Swift* e *C#*. Embora cada código gerado seja único para cada SO, a lógica de negócio (*Back-End*), permanece a mesma.

2.3.2 Aplicações Móveis Multiplataforma

Como dito por Que, Guo e Zhu (2017), para que as aplicações móveis atinjam um maior número de usuários, é necessário que esta seja desenvolvida, em pelo menos, mais de uma plataforma móvel. Os usuários de *smartphones* usam diferentes tipos de dispositivos que executam diferentes sistemas operacionais, logo, se uma aplicação quiser alcançar um maior número de usuários, ela deve ser desenvolvida para mais de um sistema operacional móvel ou ser desenvolvida para as duas plataformas mais populares *Android* e *iOS*.

Segundo Biørn-Hansen, Majchrzak e Grønli (2017), o paradigma de desenvolvimento em multiplataforma permite a construção de aplicações móveis que executam em pelo menos mais de uma plataforma móvel, a partir de um único código fonte. Já para os autores Biørn-Hansen, Majchrzak e Grønli (2017), as organizações preferem o desenvolvimento móvel multiplataforma devido ao seu custo reduzido de desenvolvimento. Eles dizem ainda que, para muitas companhias sem recursos ou sem profissionais especializados em cada plataforma móvel, o desenvolvimento em multiplataforma é modelo mais rápido e simples de se desenvolver aplicações móveis.

Todavia Que, Guo e Zhu (2017) complementam que, apesar da sua menor complexidade de desenvolvimento, as aplicações multiplataforma apresentam menor performance se comparado com as aplicações nativas. Segundo eles, isto acontece por que muitas delas são dependentes das funcionalidades disponibilizadas pelo *browser* do dispositivo.

Latif et al. (2016), Biørn-Hansen, Majchrzak e Grønli (2017) afirmam ainda que a maioria das aplicações *web* móveis multiplataforma não possuem acesso direto ao *hardware* do dispositivo. Na maioria das abordagens de desenvolvimento multiplataforma existe uma camada que intermedia as requisições entre a aplicação e o código nativo, exceto na abordagem Multi Compilada.

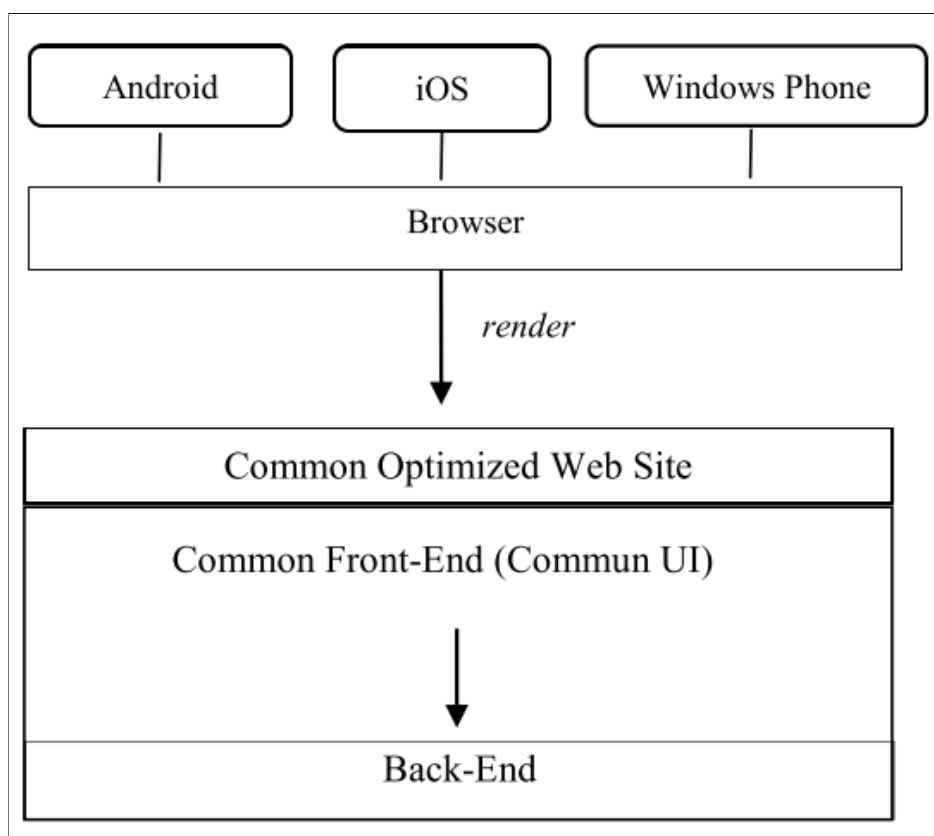
Segundo Latif et al. (2016), Latif et al. (2017), as aplicações móveis multiplataformas podem ser construídas por meio de tecnologias *web*, com ou sem junção a código nativo, interpretadores ou através de múltiplos compiladores, a partir do uso de *HTML5* e *WebViews*, mais presentes na abordagem híbrida, onde combinam código *HTML*, *CSS* e *Javascript* em um mecanismo de *web browser*, encapsulado a uma aplicação nativa.

Pode ser criado também com uso de interpretadores, que utilizam uma linguagem em comum, e assim, são interpretadas em tempo de execução por uma camada acima do SO do dispositivo. E por fim, através do uso de vários compiladores, que compilam um mesmo código em vários arquivos binários que são executados em cada plataforma móvel.

2.3.3 Aplicações Web Móveis

Em seu artigo, Jobe (2013) explica que as aplicações web móveis são projetadas para se parecer com aplicações nativas e executam em *web browsers* do dispositivo hospedeiro. Estas aplicações são construídas com base em *HTML*, *Javascript* e *CSS*. Elas podem ser comparadas a *web sites* que possuem componentes de interface adaptáveis a tela do celular. A Figura 2.7 mostra a arquitetura de uma aplicação web móvel.

Figura 2.7 – Arquitetura de uma Aplicação Web Móvel.



Fonte: Latif et al. (2016).

Como representado na Figura 2.7, uma aplicação *web móvel* pode ser criada como um *web site* ou *webApp* (PRESSMAN; MAXIM, 2016). Para isso são utilizados servidores de páginas *HTML*, *CSS* e *Javascript* e também serviços *Back-End*, como lógica do negócio e outros serviços *web*. Estes serviços são requisitados como uma requisição comum de uma página *web* que são enviados ao *browser* do dispositivo móvel. Como os *browsers* geralmente já atendem às especificações das suas plataformas nativas, a aplicação *web móvel* não precisa fazer ações adicionais de compatibilidade.

Jobe (2013) ressalta ainda que este tipo de aplicação é mais adaptada para o consumo de conteúdo. Isto acontece devido o seu projeto se assemelhar a de um *web site*. Assim, as atualizações acontecem através de novas requisições da página. Como ela é dependente do

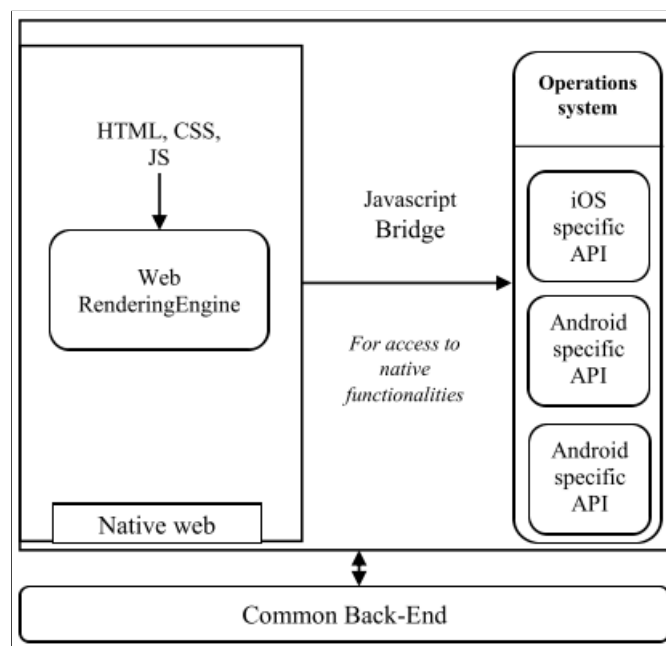
browser do dispositivo, o acesso às funcionalidades acabam sendo limitadas. Latif et al. (2017) exemplifica algumas plataformas para desenvolvimento web móvel, são elas: JQuery Mobile¹⁰, Sencha Touch¹¹ e Bootstrap¹².

2.3.4 Aplicações Móveis Híbridas

Aplicações híbridas são programadas com linguagens suportadas por *browsers* e empacotadas como código nativo (QUE; GUO; ZHU, 2017). Elas usam uma combinação entre as vantagens da abordagem de programação *web* móvel com o uso de funcionalidades nativas. O desenvolvimento híbrido utiliza-se de um mecanismo que simula um *browser* simplificado, chamado de *WebView* no Android e *UIWebView* no iOS. Estas *WebViews* encapsulam conteúdo *HTML* e acessam as funcionalidades nativas por meio de uma ponte *Javascript abstrata* (LATIF et al., 2016). A Figura 2.8 demonstra como é organizada a arquitetura de uma aplicação híbrida.

A Figura 2.8 exibe o esquema de uma arquitetura de uma aplicação móvel híbrida.

Figura 2.8 – Arquitetura de uma Aplicação Móvel Híbrida.



Fonte: Latif et al. (2016).

Na Figura 2.8 mostra o mecanismo de renderização (*Web RenderingEngine*) que recebe o código *HTML*, *CSS* e *JS* como entrada para formar a interface do aplicativo. Esse mecanismo de renderização é encapsulado por um contêiner *web* nativo, que é *WebView* ou *UIWebView*. As

¹⁰ <http://jquerymobile.com/>

¹¹ <https://www.sencha.com/products/touch/>

¹² <https://getbootstrap.com/>

funcionalidades nativas da aplicação são acessadas através de requisições feitas através de uma ponte *Javascript* (*Bridge Javascript*).

Segundo Que, Guo e Zhu (2017) as aplicações híbridas não possuem problemas com a compatibilidade entre várias versões de uma plataforma móvel, pois ela executa sobre o ambiente de execução do *browser* nativo do dispositivo, desta forma são ignoradas todas as possíveis características divergentes entre sistemas. Ele defende ainda que o desenvolvimento de aplicações híbridas economiza tempo e recursos. Aplicações híbridas desenvolvidas através de ferramentas multiplataforma tem uma maior vantagem em relação às aplicações nativas, pois, graças ao desenvolvimento em multiplataforma, elas economizam tempo e recursos.

Latif et al. (2016) ressalta que apesar dos ganhos com a economia de recursos e tempo no desenvolvimento, as aplicações híbridas perdem em performance e acesso à componentes nativos da plataforma. A sua performance é menor se comparada com a abordagem nativa, pois, a sua execução ocorre por meio de uma *WebView*, o que já custa tempo para executar uma funcionalidade.

Entre as principais plataformas de desenvolvimento desta abordagem estão: *Apache Cordova*¹³, *Ionic Framework*¹⁴, *Phonegap*¹⁵, *Angular JS*¹⁶ e *OnsenUI Framework*¹⁷ (LATIF et al., 2017).

2.3.5 Aplicações Móveis Interpretadas

A abordagem de desenvolvimento Interpretado tem como base o uso de uma linguagem comum, como por exemplo *Javascript*, que é interpretada em tempo de execução (LATIF et al., 2017). As Interfaces de Programação da Aplicação (API), presentes na plataforma móvel, são acessadas por meio de uma camada abstrata, que também é interpretada em tempo de execução. As aplicações não são dependentes de uma *WebView*, ao invés disso elas usam um interpretador presente no próprio dispositivo (BIØRN-HANSEN; MAJCHRZAK; GRØNLI, 2017).

Dentre as principais características desta abordagem, Latif et al. (2016) cita o fato de que, as aplicações criadas nesta abordagem, permitem o uso de componentes da interface nativa do dispositivo. Contudo, essa vantagem gera a dependência da aplicação no ambiente de desenvolvimento, que pode não fornecer suporte às novas ferramentas caso ocorra uma atualização da plataforma. A Figura 2.9 mostra um exemplo de uma arquitetura de uma Aplicação Móvel Interpretada.

¹³ <https://cordova.apache.org/>

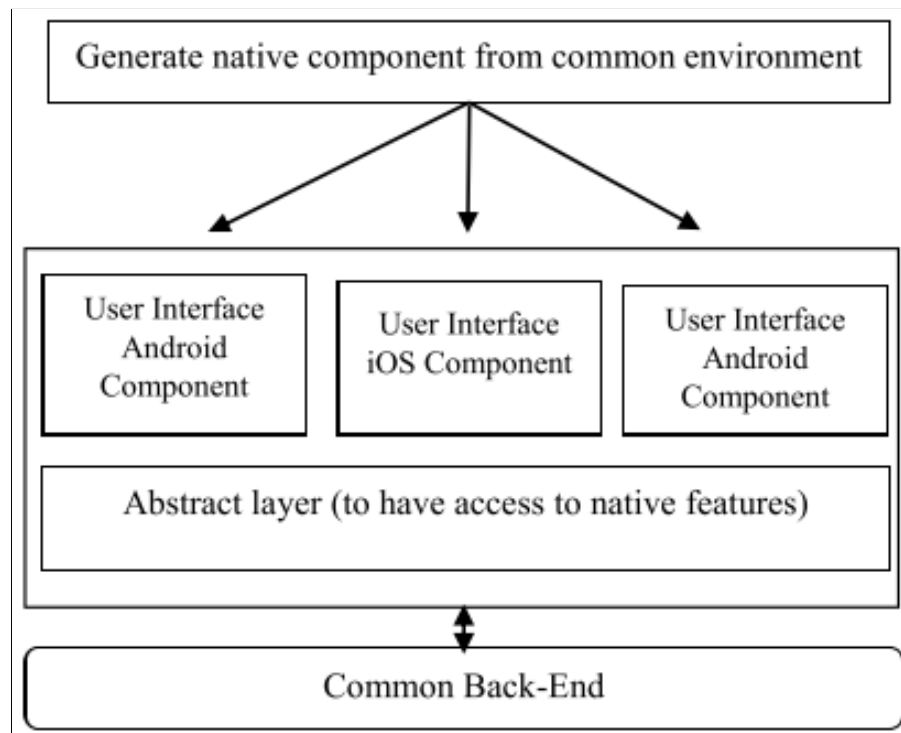
¹⁴ <https://ionicframework.com/>

¹⁵ <https://phonegap.com/>

¹⁶ <https://angularjs.org/>

¹⁷ <https://onsen.io/>

Figura 2.9 – Arquitetura de uma Aplicação Móvel Interpretada.



Fonte: Latif et al. (2016).

O modelo de arquitetura apresentado na Figura 2.9 destaca na parte superior o ambiente de desenvolvimento comum (*common environment*), responsável por gerar o código que será interpretado pelo dispositivo móvel. Ele gera os componentes de interface de usuário (*User Interface*) para cada plataforma móvel. A camada abstrata (*Abstract layer*) é responsável por fazer a ponte entre as interfaces e as APIs do SO.

Ferramentas como *Appcelerator Titanium*¹⁸, *Smartface App Studio*¹⁹ e *React Native*²⁰ são as mais comuns entre os desenvolvedores desta abordagem (LATIF et al., 2017).

2.3.6 Aplicações Móveis Multi Compiladas

Na abordagem de aplicações móveis Multi Compiladas, o código fonte é compilado em diversos compiladores de uma só vez, para cada plataforma móvel para ser mais exato, gerando um código executável adaptado ao SO de cada plataforma (LATIF et al., 2016).

Latif et al. (2017) complementa ainda que, as maiores vantagens desta abordagem é que ela fornece uma performance muito próxima a das aplicações nativas e possui acesso aos

¹⁸ <https://www.appcelerator.com/>

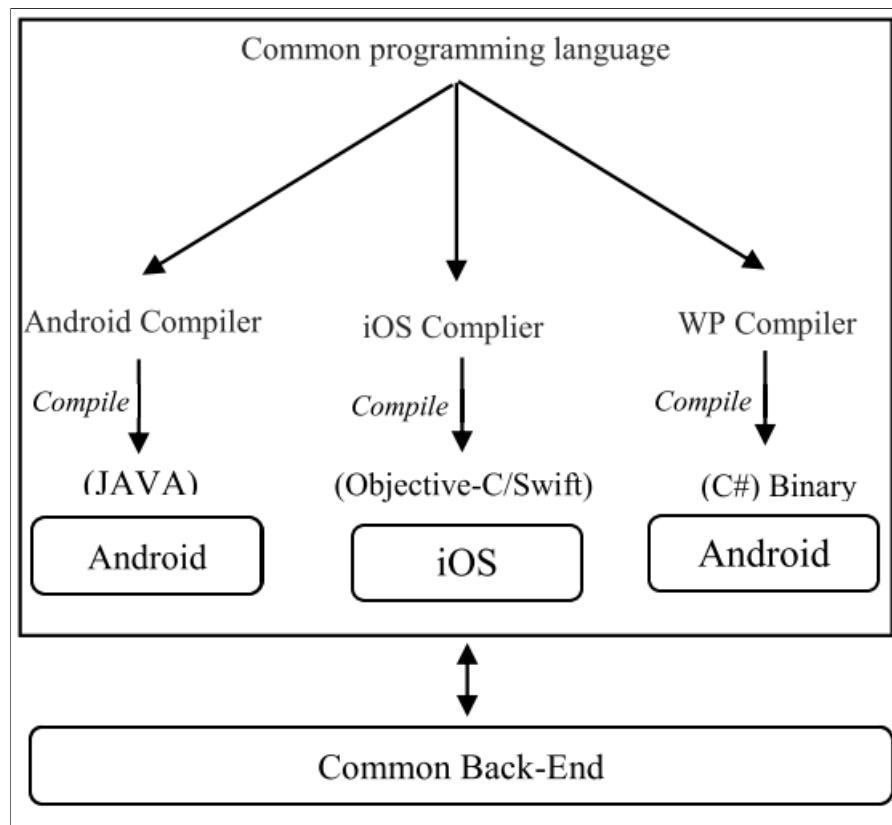
¹⁹ <https://www.smartface.io/smartface/>

²⁰ <https://facebook.github.io/react-native/>

componentes de interface nativas. Contudo, sua maior desvantagem é a pouca disponibilidade de ferramentas que ela pode acessar, como por exemplo, o GPS ou câmera do celular. Isto ocorre porque estas ferramentas são específicas de dispositivo para dispositivo.

A Figura 2.10 mostra como ocorre a construção de uma aplicação multiplataforma.

Figura 2.10 – Arquitetura de uma aplicação móvel Multi Compilada.



Fonte: Latif et al. (2016).

Na parte superior da Figura 2.10, observa-se o uso de uma linguagem de programação comum (*Common programming language*) para a criação do código fonte. Este código por sua vez é compilado em diversos compiladores, como *Android Compiler*, *iOS compiler* e *WP Compiler*. O código gerado por cada um dos compiladores pode ser instalado em suas respectivas plataformas como uma aplicação puramente nativa, apesar de ambos compartilharem de uma mesma lógica de negócio (*Common Back-End*).

Exemplos de ferramentas de desenvolvimento que se utilizam desta abordagem são: *Xamarin*²¹ e *CodeName One*²² (LATIF et al., 2017).

²¹ <https://www.xamarin.com/>

²² <https://www.codenameone.com/>

2.4 Android

Como foi dito por Simões e Pereira (2014) e Pressman e Maxim (2016), o *Android* é uma plataforma de software que está presente em diversos aparelhos móveis, sejam eles *smartphones*, *tablets*, *smartwatches*, carros, aparelhos domésticos e muitos outros. Ele é construído em cima do *Kernel* do *Linux*, que é a base para uma Camada de Abstração de Hardware, que em inglês se chama *Hardware Abstraction Layer (HAL)*. A HAL mantém as interfaces de *hardwares* como *bluetooth*, câmera, sensores e outros componentes, que são utilizados pelas Bibliotecas Nativas C/C++ e pela *Android Runtime*. A *Android Runtime* depende de das bibliotecas C/C++ e fornece serviços para a camada do *Java API Framework*, que é onde reside as bibliotecas codificadas em *Java* que implementam o acesso à estes serviços. E por fim, no topo, encontram-se as aplicações móveis que fazem o uso da *Java API Framework* (GOOGLE DEVELOPERS, 2018).

As características de cada camada e os serviços que cada uma oferece são detalhadas dolo abaixo segundo GOOGLE DEVELOPERS (2018):

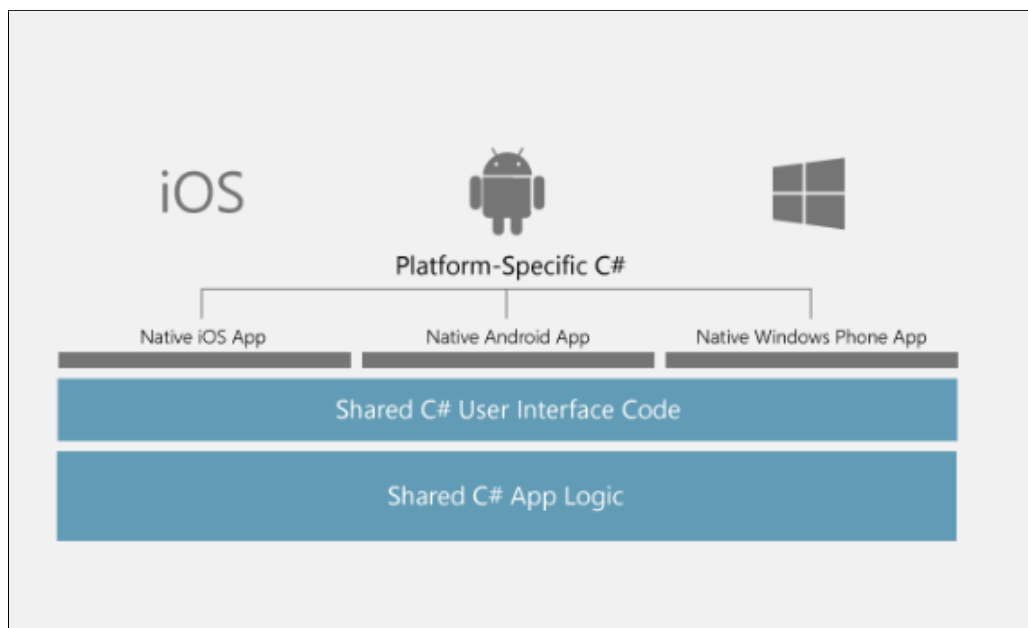
- **Linux Kernel:** é a fundação da plataforma Android. Ela lida com funcionalidades a nível de Sistema Operacional como encadeamento e gerenciamento de memória. Ela é muito utilizada pela *Android Runtime*;
- **Hardware Abstraction Layer (HAL):** nesta camada são guardadas as *interfaces* que acessam os módulos de *hardwares* do dispositivo como o módulo de câmera ou *bluetooth*. Quando uma biblioteca *Java* faz uma chamada para acessar um hardware, o sistema *Android* carrega um módulo desses módulos e os acessa por meio de uma *interface* correspondente, mantida na HAL;
- **Android Runtime (ART):** É responsável por gerenciar os processos das aplicações em seu tempo de execução. Ela foi projetada para executar várias máquinas virtuais em dispositivos de baixa memória. Além disso ela permite a compilação *Ahead-of-Time* (AOT) e também a *Just-in-Time* (JIT), coleta de lixo otimizada e depuração otimizada;
- **Native C/C++ Libraries:** vários componentes necessários à ART e a HAL são implementados por código nativo e exige bibliotecas nativas programadas em C e C++. Como por exemplo, é possível acessar *OpenGL ES* pela *Java OpenGL API* da estrutura do Android para adicionar a capacidade de desenhar e manipular gráficos 2D e 3D no aplicativo;
- **Java API Framework:** São bibliotecas construídas com código Java que podem ser utilizadas em aplicativos. Elas simplificam o processo de implementação de componentes de interface de usuário e oferecem serviços de gerenciamento de componentes do sistema;

- **System Apps:** nesta camada estão os aplicativos desenvolvidos para o *Android*. Existem aplicações de *Email*, *Camera*, *Calendário* e outros. Além dos aplicativos fornecidos pela própria *Google*, é possível desenvolver aplicações próprias por meio do SDK do *Android*.

2.5 Xamarin

Xamarin é uma plataforma de desenvolvimento voltada para aplicações móveis mantida pela *Microsoft* e pela *.NET Foundation* (MONOPROJECT, 2018). Ela é desenvolvida sobre o *Mono* que é um *framework* para desenvolvimento em multiplataforma. O *Xamarin* implementa o paradigma de desenvolvimento Multi Compilado através da linguagem comum C# ou F#. Ele utiliza *Visual Studio*²³ como seu *Integrated Development Environment* (IDE). O *Xamarin* fornece bibliotecas de código escritos em C# que são compilados e mapeados para uma linguagem intermediária, depois este código é compilado para o código nativo. A Figura 2.11 mostra a arquitetura de uma aplicação desenvolvida com o *Xamarin*.

Figura 2.11 – Arquitetura de uma aplicação desenvolvida no Xamarin.



Fonte: Microsoft (2018).

Como é mostrado na Figura 2.11, aplicação compartilha código C da camada de lógica de negócio (na figura é chamada de *App Logic*) e na camada de interface do usuário. Nesta abordagem é empregado o um subproduto do *Xamarin* que é chamado de *Xamarin.Forms*, que nada mais é que um framework de componentes de UIs *Xamarin Native* e o *Xamarin.Forms*

²³ <https://visualstudio.microsoft.com/pt-br/?rr=https%3A%2F%2Fwww.google.com.br%2F>

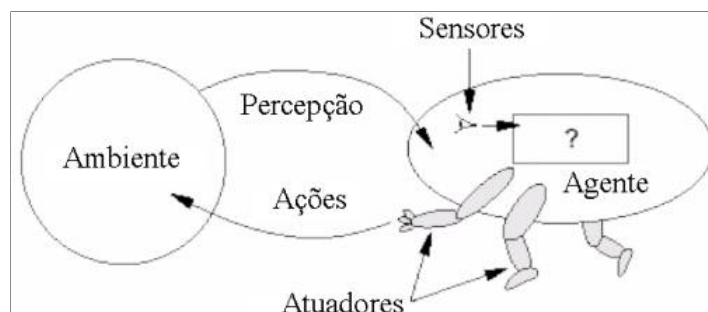
(MICROSOFT, 2014). O *Xamarin.Forms*²⁴ Consiste em um *framework* de interfaces de usuários que possui comportamentos semelhantes em todas as plataformas. Mas ainda sim existe um outra abordagem de construção de interfaces que utiliza componentes nativos da plataforma traduzidos para código C#, esta abordagem é chamada de *Xamarin Native*. No *Xamarin Native* é possível programar componentes de interfaces de usuário nativas da plataforma, mas esta abordagem não permite a reutilização de código para as outras plataformas. Para esta abordagem foram desenvolvidos dois *frameworks*: *Xamarin.Android*²⁵ e *Xamarin.iOS*²⁶. Nesta abordagem é necessário o conhecimento dos conceitos de cada plataforma, pois, o *Xamarin* mapeia o código C para a linguagem da plataforma e assim, utiliza o compilador da mesma para gerar o respectivo arquivo binário.

2.6 Agentes de Softwares

A evolução dos softwares e hardwares, acompanhada do surgimento de novos meios de coleta de dados através de uso de sensores, vem exigindo novas formas de abstrações de software que se aproximem mais dos objetos reais, tanto em comportamento, quanto em autonomia e interação com o usuário (SUN; LI; XIE, 2013; RUSSELL; DOYLE; COLLIER, 2018). O projeto de agentes de software traz esta nova forma de abstrair objetos do mundo real, permitindo que o software se comporte de forma autônoma e tome decisões sem a necessidade da intervenção direta do usuário (JEZIC et al., 2018).

Russell et al. (2014) definem um agente como qualquer coisa que observa o ambiente a que está inserido, por meio de sensores e executa ações sobre este ambiente através de seus atuadores. A Figura 2.12 mostra a representação de um modelo de agente criado por Russell et al. (2014).

Figura 2.12 – Modelo conceitual de um agente.



Fonte: Oocities.org (2018).

²⁴ <https://docs.microsoft.com/pt-br/xamarin/pivot=platformspanel=XamarinForms>

²⁵ <https://docs.microsoft.com/pt-br/xamarin/android/>

²⁶ <https://docs.microsoft.com/pt-br/xamarin/ios/>

Como mostra a Figura 2.12, o agente percebe as mudanças que ocorrem em seu ambiente através dos sensores, se alguma percepção estiver mapeada à alguma ação, os atuadores executam esta ação que modifica o seu ambiente. Russell et al. (2014) assemelham o ser humano a um agente, devido ao fato de que este possui olhos, ouvidos e nariz que são os sensores, enquanto que suas mãos, pés, boca e outras partes do corpo, são os seus atuadores. O ser humano age de acordo com as percepções que são capturadas pelos seus sensores, que por vez modifica o ambiente através dos seus órgãos atuadores. Agentes robóticos podem usar câmeras como sensores e motores como atuadores. Um agente de software pode perceber o ambiente através de strings e agir por meio delas.

Leitão e Karnouskos (2015) conceituam um agente como uma entidade computacional autônoma, solucionadora de problemas e orientada a objetivos, com habilidades sociais que são capazes de atuar, seja de forma proativa ou não, sobre um ambiente dinâmico e aberto, no sentido de observar e agir para alcançar os seus objetivos.

Já Belghiat et al. (2016) consideram um agente como uma entidade de software que executa tarefas em nome de alguém (uma pessoa), organização ou outro agente, com alguma autonomia, isto é, suas ações podem não ser somente determinadas por eventos externos ou interações, mas também por sua própria motivação.

Tomando como base os conceitos citados anteriormente, um agente de software é um programa projetado sobre uma arquitetura que lhe permita perceber mudanças em um ambiente computacional, com uma certa autonomia, com base em objetivos que são mapeados em ações. Estas ações são executadas por meio de atuadores que podem ser componentes físicos ou virtuais.

2.6.1 Tipos de Agentes de Software

Segundo Leitão e Karnouskos (2015), os agentes podem ser classificados quanto ao nível de autonomia e inteligência que é empregado em sua construção, que vai de agentes reativos até os agentes deliberativos.

Os agentes deliberativos são aqueles que aplicam os conceitos de representação simbólica com semânticas compostas, como por exemplo, uso de árvore de dados. Sua arquitetura costuma ser mais complexa, pois ele não somente age sobre os fatos, como também, gera teorias e analisa o passado para obter a melhor forma de resolução de um problema. O seu ponto fraco baseia-se nas consequências naturais do seu comportamento, que são o alto consumo de recursos, lentidão na tomada de decisões e alta complexidade do projeto (LEITÃO; KARNOUSKOS, 2015).

Já os agentes reativos possuem uma arquitetura mais simples por não precisar de estruturas complexas de raciocínio. Sua performance é maior e não consome tantos recursos como os agentes deliberados. Esta maior performance se deve ao modelo de decisão que é feito através de uma simples associação entre uma situação e uma ação, normalmente implementados por

um padrão de *Mapping*. Em contrapartida, não são flexíveis e dinâmicos na resolução de problemas desconhecidos, como também, não são proativos e dificilmente são orientados a objetivos (LEITÃO; KARNOUSKOS, 2015).

Quanto a forma de construção da arquitetura de um programa agente, segundo Russell et al. (2014), ele pode ser construído de quatro formas diferentes:

- **Agentes Simples Reflexivos:** neste tipo de arquitetura o agente possui os componentes básicos necessários para o seu funcionamento. Ele mapeia as percepções às ações por meio de estruturas simples de decisão. Se utiliza apenas de sensores, atuadores e uma estrutura de decisão simples;
- **Agentes Reflexivos com Estados:** esta arquitetura é projetada para manter o estado do ambiente após a execução de uma ação sobre o ambiente. Este modelo é ideal para agentes que precisam fornecer ações em ambientes instáveis. Além dos sensores, atuadores e estrutura de decisão, é também implantado um componente que armazena os estados do ambiente;
- **Agentes Reflexivos com Objetivos:** este modelo é a evolução de um Agente Reflexivo com estado que se preocupa com as consequências de suas ações e possui um componente que reúne os objetivos do agente;
- **Agentes baseados em Utilitários:** esta arquitetura adiciona um componente de Utilidade ao modelo de arquitetura de um Agente Reflexivo com Objetivos. Este componente define um valor de prioridade para auxiliar na escolha do melhor objetivo para o agente.

Neste trabalho foi criado uma assistente virtual através da construção de um agente de software reativo, que agrupa os componentes presentes em um modelo de Agente Reflexivo com Estados.

2.7 Desenvolvimento de Software

Sobre o desenvolvimento de software, Pressman e Maxim (2016) falam que no início os softwares eram mais simples e possuíam menor custo para serem criados, a ponto de ser desenvolvido por apenas um programador. Com o passar dos anos esses softwares foram evoluindo, se tornando cada vez mais complexos, demandando mais tempo e recursos para serem construídos. Desta forma, o que antes era feito por apenas um programador, acabou necessitando de uma equipe bem disciplinada para poder fazê-lo. Esta nova era instaurou o caos dentro do desenvolvimento, pois, eram muitas pessoas trabalhando de forma desordenada, sem nenhum controle sobre o tempo e custo do software. Para resolver este problema, foram criados modelos

de processos de desenvolvimento de software, justamente para padronizar todo o desenvolvimento e sincronizar os custos e prazos do software a ser entregue. O Processo Unificado da Rational (RUP) é um exemplo de processo de desenvolvimento de software que foi utilizado na construção do aplicativo tema deste trabalho, ele é abordado nos próximos subtópicos.

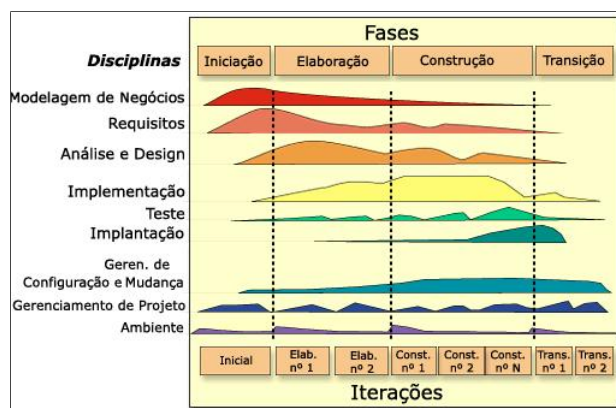
2.7.1 RUP

Conhecido primeiramente como PU (*Process Unified*), foi criado por *Ivar Jacobson*, *Grady Booch* e *James Rumbaugh*. Logo depois fora conhecida pelo nome de *Rational Unified Process* em homenagem à *Rational Corporation*, uma das primeiras colaboradoras para o desenvolvimento do PU, que mais tarde fora adquirida pela IBM (PRESSMAN; MAXIM, 2016).

Jin e Liang (2016) fala que o RUP foi criado com o intuito de ser um processo de desenvolvimento de software dirigido a casos de uso, centrado na arquitetura, iterativo e incremental. É dirigido a casos de usos, pois, é através deles que são identificados todos os atores e envolvidos, além de delimitar as funcionalidades do software. Possui um enfoque na construção de uma arquitetura executável, que seja flexível e resiliente a mudanças, uma vez que este é o ponto chave da vida do software, já que deverá suportar a inserção de vários componentes e ainda manter-se confiável e seguro. A iteração é nada mais que o ciclo completo do processo de desenvolvimento que pode ser repetido várias e várias vezes, culminando ao término de cada ciclo no lançamento de uma nova versão do software. E por fim, incremental devido a natureza do software ser muito complexa, a ponto de ser dividido em partes menores para serem construídas. Ao final de cada iteração um novo incremento do software é adicionado a arquitetura, resultando em um novo software mais completo.

O RUP é um processo dividido em quatro fases: Iniciação/Concepção, Elaboração, Construção e Transição. A Figura 2.13 descreve as fases do RUP e as relaciona com as suas respectivas atividades genéricas.

Figura 2.13 – Fases do RUP.



Fonte: Adaptado de (SOMMERVILLE, 2011).

As fases presentes na Figura 2.13 são descritas a seguir, segundo a especificação da IBM (2004):

- **Iniciação/Concepção:** Nesta fase, identificam-se os casos de uso do sistema, seus atores e é delimitado o escopo do sistema. Nesta etapa também são analisados os riscos do projeto, estimativas de recursos necessários e planejamento dos incrementos e iterações que serão implementados. Nesta etapa um maior esforço é gasto nas atividades de modelagem de negócio e de elicitação de requisitos de software;
- **Elaboração:** Esta fase é responsável por preparar a arquitetura do software para receber os componentes. Neste ponto a deve ser criado um protótipo executável da arquitetura, onde ele será complementado a cada iteração. Esta fase compreende um maior esforço nas atividades de análise e projeto da arquitetura do software, como também na implementação, neste caso, da arquitetura;
- **Construção:** Nesta etapa são construídos e testados os componentes que serão vinculados a arquitetura do software. Nesta etapa também são criadas métricas que auxiliam no controle do tempo e dos recursos. Durante esta etapa, as atividades de implementação e testes são mais atuantes e devem dar início as atividades de implantação;
- **Transição:** Etapa final onde o software é preparado para ser disponibilizado aos seus usuários ou é lançada uma nova versão. Ao chegar neste ponto, o software estará se preparando para lançar uma nova versão, por isso, a atividade de implantação possui maior prioridade.

Segundo Pressman e Maxim (2016), *James Rumbaugh, Grady Booch e Ivar Jacobson* começaram então a trabalhar em um conjunto de modelos que representassem melhor os conceitos de projetos Orientados a Objetos em cada uma das atividades de cada fase do RUP. Foi aí que surgiu a *UML (Unified Modeling Language)*, que contém uma notação robusta para modelagem e desenvolvimento de sistemas orientados a objetos. Seus modelos foram utilizados nas suas respectivas atividades dentro de cada fase deste projeto.

2.7.2 Processo de desenvolvimento de Agentes de Software para dispositivos móveis

O modelo de processo *RUP* é utilizado como um modelo de apoio na construção de um Sistema Baseado em Agentes (SBA) para dispositivos móveis (DINSOREANU; SALOMIE;

PUSZTAI, 2002). Pressman e Maxim (2016) definem um Processo como sendo um conjunto de atividades metodológicas, ações e tarefas realizadas na criação de algum artefato. Segundo Cheah et al. (2017) o desenvolvimento de SBAs possui as suas próprias atividades metodológicas, tarefas e ações que resultam em artefatos próprios para sua construção. Neste trabalho estas atividades metodológicas foram organizadas dentro das fases do *RUP* que compreendem conceitos semelhantes. O Quadro 2.1 exibe algumas atividades metodológicas, que por sua vez, são compostas por tarefas propostas por pesquisadores especializados no desenvolvimento de SBAs.

Quadro 2.1 – Atividades metodológicas e tarefas para o desenvolvimento de um SBA.

Atividades	Tarefas	Autor(es)
Elicitação de Requisitos	Comunicação com interessados. Revisão de Literatura. Busca por sistemas semelhantes. Construção de modelos <i>RO-ADMAP</i> e <i>i*</i> Objetivos. Elicitação de requisitos Funcionais e Não-Funcionais. Validação dos requisitos.	Cheah et al. (2017). Badham et al. (2018).
Análise e Projeto de Arquitetura	Gerar Diagrama de Pacotes (UML). Gerar Diagrama de Use case (UML). Gerar Diagrama de Classe (UML). Gerar Modelo de Máquina de Estados (UML). Gerar Modelo de Sequência (UML).	Dinsoreanu, Salomie e Pusztai (2002). Guedes e Vicari (2010). Marco e Pace (2013).
Construção e Testes	Construir Agente. Implementar casos de testes.	Marco e Pace (2013). Russell, Doyle e Collier (2018).
Implantação e Publicação	Gerenciar versionamento. Gerenciar lançamento de versões <i>Release</i> (Interna, Alfa, Beta e Produção).	Sommerville (2011). Pressman e Maxim (2016). Russell, Doyle e Collier (2018).

Fonte: Autor.

Neste trabalho foram selecionadas as tarefas do Quadro 2.1 para auxiliar a construção do SBA. As atividades de Elicitação e Requisitos foram incluídas a fase de Concepção do *RUP*.

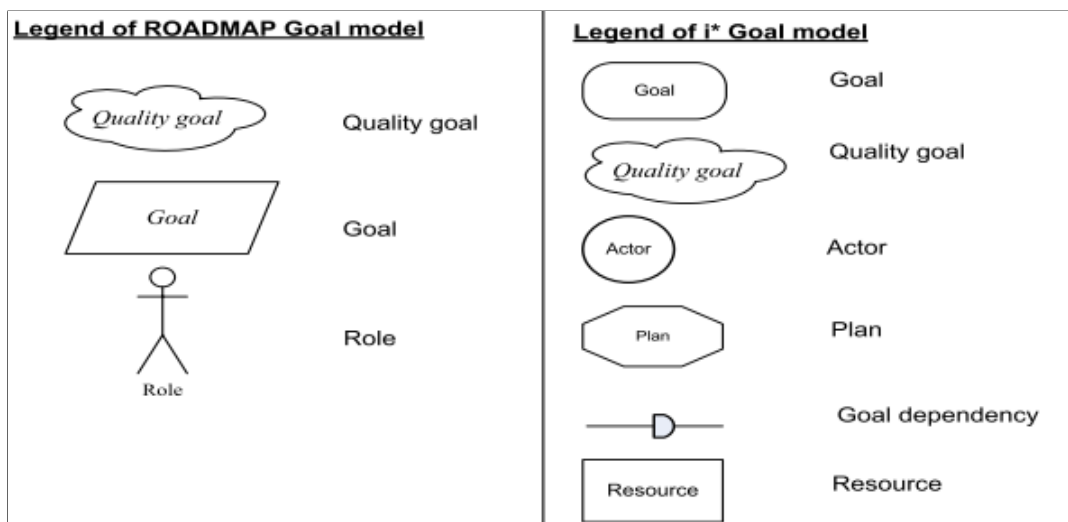
As atividades de Análise e Projeto de Arquitetura foram incorporadas à fase de Elaboração. Dentro da fase de Construção, foram inseridas atividades de Construção e Testes. E por fim, as atividades de Implantação e Publicação são mapeadas para a fase de Transição do *RUP*.

2.7.3 ROADMAP e modelos de i* Objetivos

Segundo (SOMMERVILLE, 2011), a etapa de Engenharia de Requisitos consiste em definir o que o sistema deverá fazer, os serviços que ele oferece e as restrições de suas funcionalidades. (BADHAM et al., 2018) falam que tanto SBAs como também Sistemas Centrados em Agentes (SCA), são tipos de sistemas se adaptam sobre o tempo, a medida que decisões vão sendo tomadas e mudanças vão ocorrendo no ambiente. Estas decisões são tomadas através das regras que são programadas dentro deles. Então, a Elicitação de Requisitos acaba sendo focada na identificação destas regras, do ambiente em que elas serão executadas e dos recursos que elas manipularão ou são dependentes.

Para auxiliar na Elicitação de Requisitos de SBAs, foi utilizada a técnica proposta por Cheah et al. (2017), que consiste na elaboração de dois modelos orientado a objetivos, chamados de *ROADMAP* e *i* goal modelling* (modelos de i* objetivos). O *ROADMAP* é um modelo de alto nível que mostra as necessidades dos usuários do SBA e representam também suas funções e características visíveis ao usuário. Os modelos de i* objetivos é usado para detalhar cada objetivo identificado pelo *ROADMAP* mostrando as funcionalidades do sistema, suas dependências e requisitos não-funcionais. A Figura 2.14 mostra a notação de cada um dos modelos.

Figura 2.14 – Notação dos modelos orientados à objetivos sugerido por (CHEAH et al., 2017).

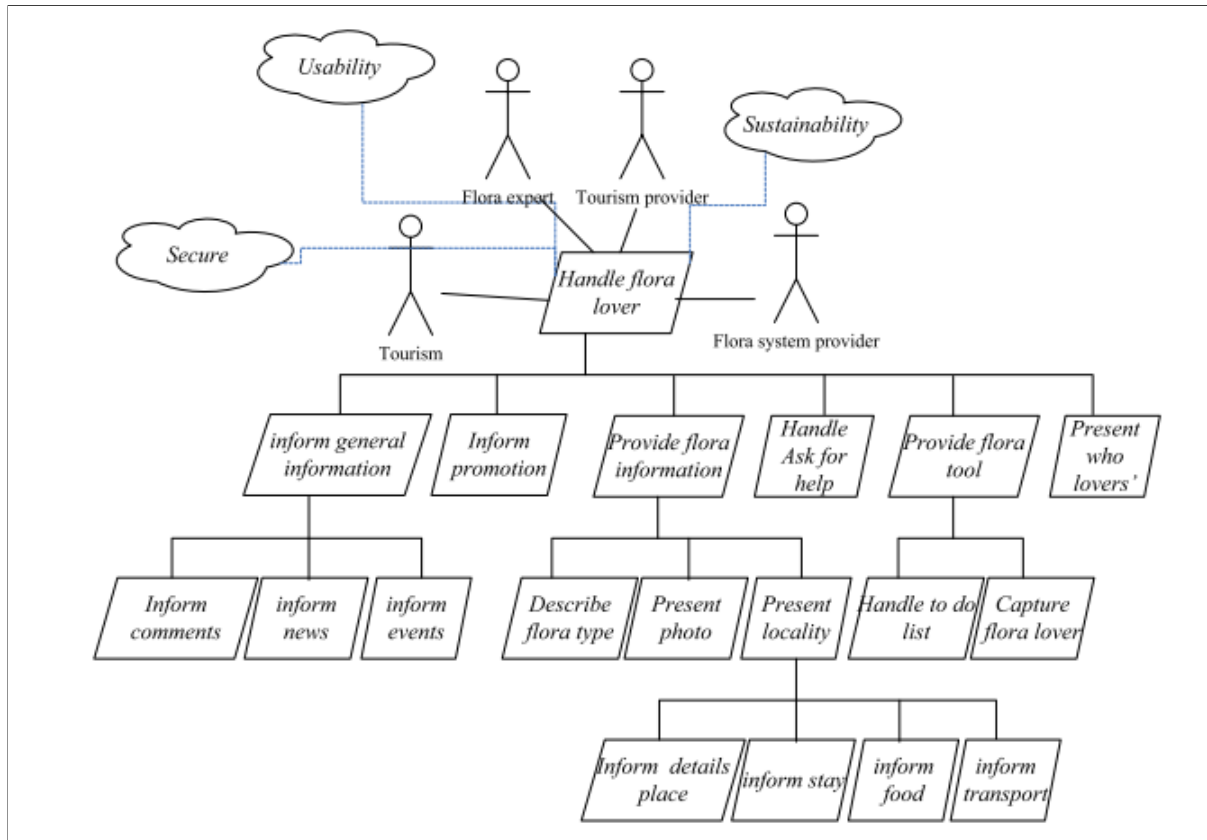


Fonte: Cheah et al. (2017)

A Figura 2.14 mostra a notação do *ROADMAP* à esquerda e a notação dos i* modelos de objetivos, que se encontra à direita. O *ROADMAP* modelará os papéis, objetivos, subobjetivos,

papéis, objetivos de qualidade e os relacionamentos entre os papéis e os objetivos. Já os modelos de *i** objetivos modela o sistema com objetivos, subobjetivos, dependências dos objetivos, atores, planos e recursos. Um exemplo de um *ROADMAP* é apresentado na Figura 2.15.

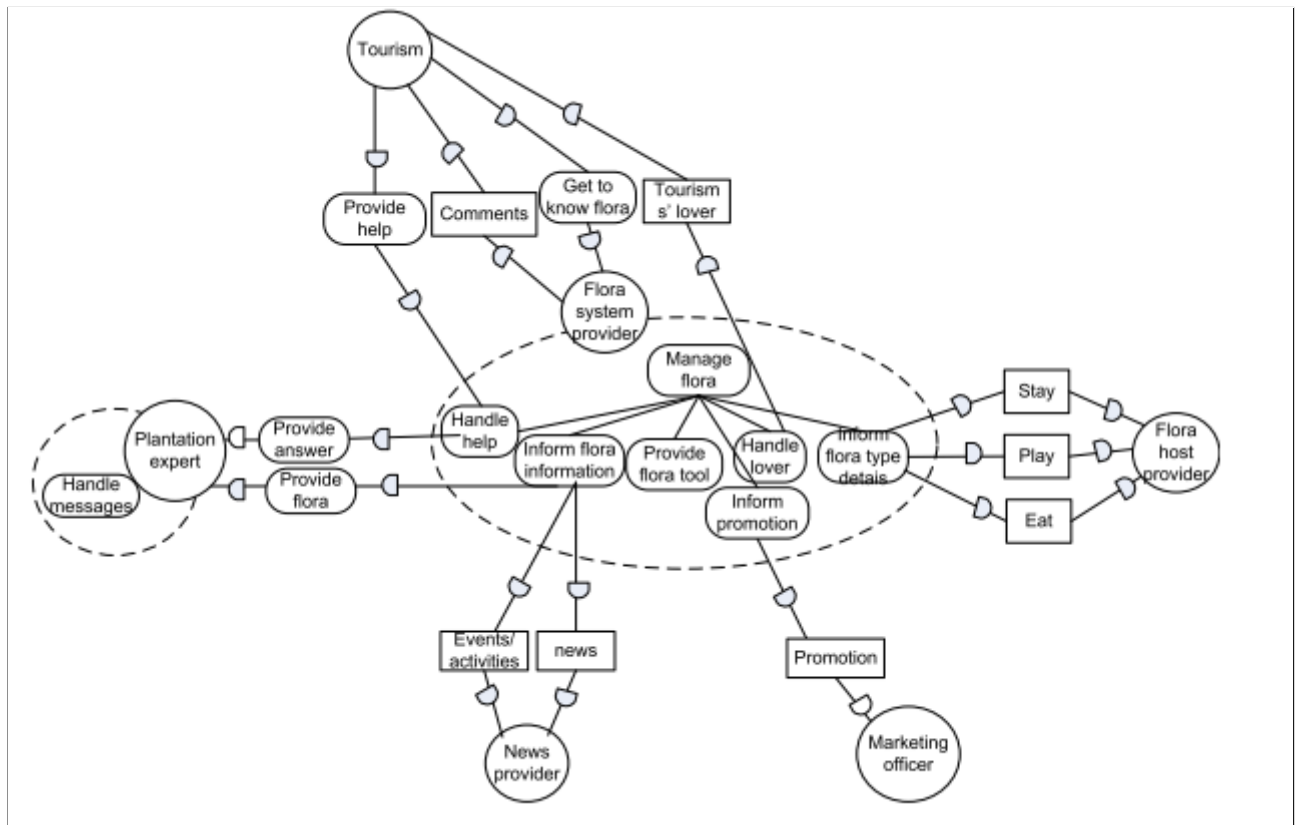
Figura 2.15 – Exemplo de uma modelagem com *ROADMAP*.



Fonte: Cheah et al. (2017)

Na Figura 2.15, mostra um modelo feito em *ROADMAP* representando um aplicativo chamado de *Kuching FloraLover*. Uma aplicação móvel voltada para a popularização do ecoturismo da cidade de *Kuching*, em *Sarawak, Malásia*. Seu objetivo é promover o turismo da biodiversidade de *Sarawak*. Ele armazena informações e a localização de locais de visitaç o de florestas nativas, isto  , eventos, atividades na flora, jardim, etc. Ele foi projetado para atrair amantes de plantas.

A Figura 2.15 aponta os seguintes atores: *Tourism*, *Flora expert*, *Flora System Provider* e *Tourism Provider*. Ambos se relacionam com o objetivo "*Handle flora lover*" que representa a funcionalidade principal do sistema. E logo abaixo dela, s o representados os seus subobjetivos. Depois cada objetivo e subobjetivo   detalhado por meio do modelo *i** objetivos. A Figura 2.16 exibe o modelo *i** objetivo do *Kuching FloraLover*.

Figura 2.16 – Modelo de i* objetivos do aplicativo *Kuching FloraLover*.

Fonte: Cheah et al. (2017).

A Figura 2.16 mostra as dependências entre os atores, objetivos e recursos necessários para a execução das tarefas do agente do aplicativo *Kuching FloraLover*. Como por exemplo, o ator *Tourism*, para atingir o objetivo "Provide help", vai depender do objetivo do "Handle help" que por sua vez se relaciona com o objetivo do "Manage flora" do agente "Flora System Provider". O círculo tracejado, representa o domínio do ator que se encontra sobre ele.

Ao final da construção dos modelos *ROADMAP* e o de i* objetivos, os objetivos são transformados em Requisitos Funcionais do sistema e os objetivos de qualidade serão transformados em Requisitos Não-Funcionais.

2.8 Considerações do capítulo

Neste capítulo definiu-se a doença *Diabetes Mellitus* (DM) como sendo uma doença crônica e grave que atinge milhares de pessoas pelo mundo. Ela é caracterizada por problemas com o uso de insulina pelo corpo, o que causa o acúmulo de glicose no sangue responsável por criar várias complicações de saúde. Ela pode ser classificada em DM tipo 1, DM tipo 2 ou DM gestacional, sendo que a maioria dos DM são do tipo 2. Seu tratamento consiste no controle dos níveis de glicose no sangue que é dependente de um modo de vida saudável. Esse

tratamento pode ser melhorado através da inserção de tarefas de autocuidado ao qual permitem que o paciente viva um estilo de vida saudável e de qualidade. Dentre todas as tecnologias para o autocuidado, encontram-se os aplicativos móveis que, devido a sua popularidade e baixo custo, são uma ótima ferramenta para implantar mudanças de comportamento quanto ao tratamento do DM. Foi abordado sobre o *Android*, demonstrando toda a sua arquitetura e funcionamento. Logo em seguida, foi mostrado o *framework Xamarin* para desenvolvimento de aplicações móveis Multi Compiladas, que foi o *framework* de desenvolvimento utilizado para a construção do aplicativo. Para justificar o uso de uma arquitetura centrada em Agentes, foi mostrado os conceitos relacionados à Agentes de software, como também, foi apresentado os tipos de Agentes e suas características. E por fim, foi descrito o modelo de processo de desenvolvimento utilizado no projeto do aplicativo, o *RUP*, descrevendo cada uma das suas fases e atividades. Foi mostrado como o *RUP* encapsulou as atividades metodológicas e tarefas para a construção de um Sistema Baseado em Agente para aplicações móveis, onde foi falado sobre os modelos conceituais *ROADMAP* e modelos de *i** Objetivos. O próximo capítulo descreve a pesquisa de campo executada neste trabalho.

3 Pesquisa de campo

Executou-se uma pesquisa de campo de cunho exploratório e quantitativo com o objetivo principal de levantar informações sobre o uso de aplicativos móveis por pessoas portadoras de Diabetes *Mellitus*. A pesquisa teve como objetivos secundários reunir as principais dificuldades no uso dos aplicativos, quais funcionalidades mais importantes para os pacientes e avaliar o nível de interesse no projeto de uma solução de software para *smartphones*.

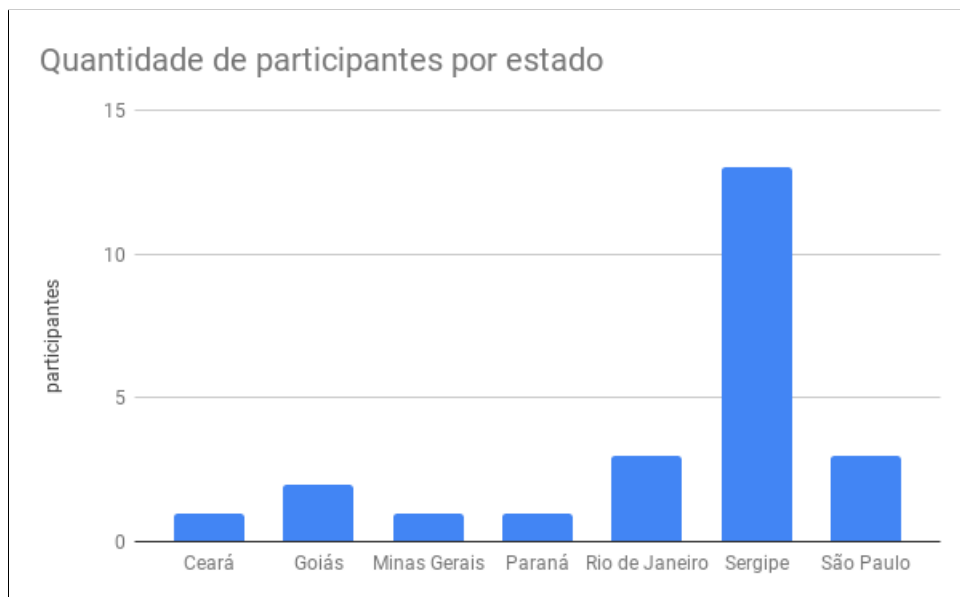
Esta pesquisa foi executada por meio da aplicação de um questionário disponibilizado pela internet durante o período de 14/02/2018 até 20/04/2018. O formulário se encontra no Apêndice 7 ao final deste trabalho. Participaram da pesquisa 24 pessoas de diferentes faixas etárias e estados do Brasil que são portadoras de Diabetes *Mellitus* tipo 1, tipo2, gestacional ou que estão dentro do estágio de pré-diabetes.

Foram colhidas informações quanto a renda familiar, acesso a *smartphones*, frequência de utilização *smarphones*, se utiliza aplicativos para o cuidado no diabetes, se sentem dificuldades ao utilizar os aplicativos, qual funcionalidade são mais importantes nos aplicativos, validação de uma solução de *software* e se estariam dispostos a investir no aplicativo Tia Bete.

3.1 Resultados da pesquisa

O gráfico da Figura 3.1 exibe a quantidade de participantes da pesquisa por estado.

Figura 3.1 – Gráfico do percentual de participantes por estado.



Fonte: Autor.

Foram encontradas informações dos estados Ceará, Goiás, Minas Gerais, Paraná, Rio de Janeiro, Sergipe e São Paulo. Como disposto na Figura 3.1, Sergipe foi o estado com mais participantes na pesquisa com 13 pessoas, logo em seguida estão os estados de São Paulo e Rio de Janeiro com 3 participantes para ambos.

Quanto a faixa etária, o gráfico da Figura 3.2 mostra a distribuição participantes na pesquisa.

Figura 3.2 – Gráfico dos participantes por faixa etária.



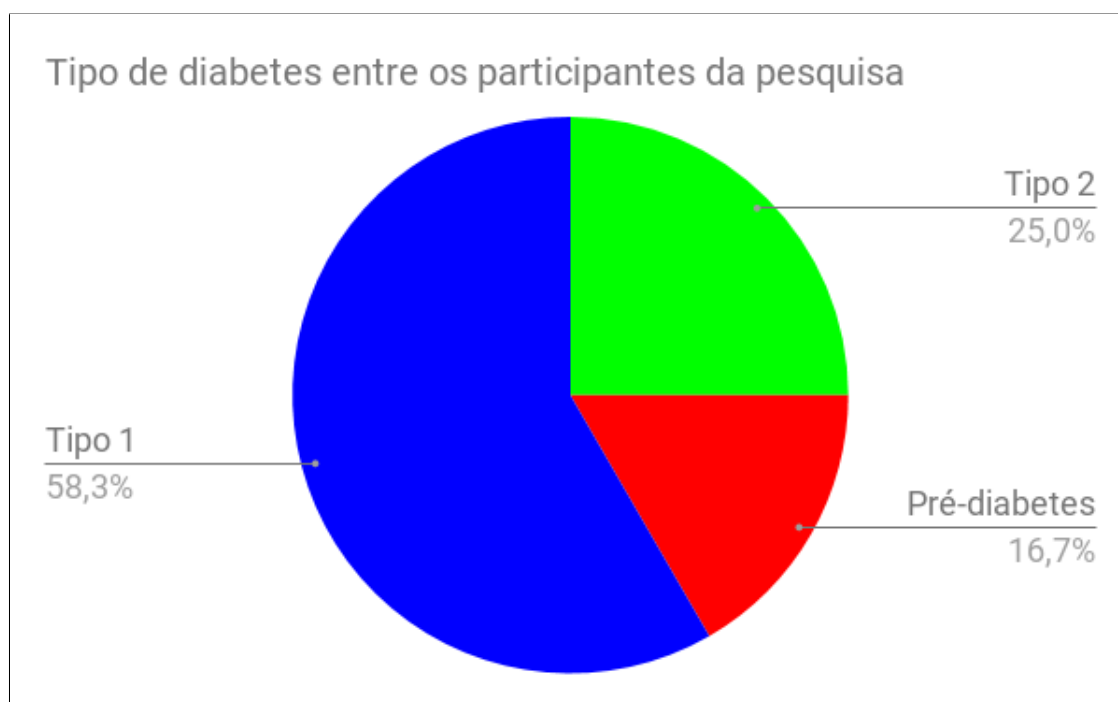
Fonte: Autor.

Como mostra a Figura 3.2 a maioria dos participantes eram menores de 20 anos de idade. Isto tende a acontecer devido o fato de que as pessoas que mais acessam a internet (IBGE, 2016). Em seguida as pessoas entre 40 e 49 anos com 6 pessoas. Foram encontradas 2 pessoas na pesquisa com idade acima dos 59 anos e entre 20 e 29 anos, que foram os menores resultados. Sendo que 33,3% deles eram homens e 66,7% eram mulheres.

Quanto a renda familiar, a maior parte (54,2%) relatou ganhar acima de 5.000,00 R\$. Não houveram pessoas que possuam renda familiar entre R\$4.000,00 a R\$4.900,00. De R\$3.000,00 a R\$3.900,00 foi encontrada 1 pessoa, o mesmo se sucedeu para a faixa de R\$2.000,00 a R\$2.900,00. 25% dos entrevistados têm renda familiar de Cerca de 25% declaram ter de um salário-mínimo a R\$1.900,00 e 12% menos que um salário-mínimo.

A Figura 3.3 mostra a disposição dos pacientes quanto ao tipo de diabetes.

Figura 3.3 – Gráfico do Tipo do diabetes entre os participantes.

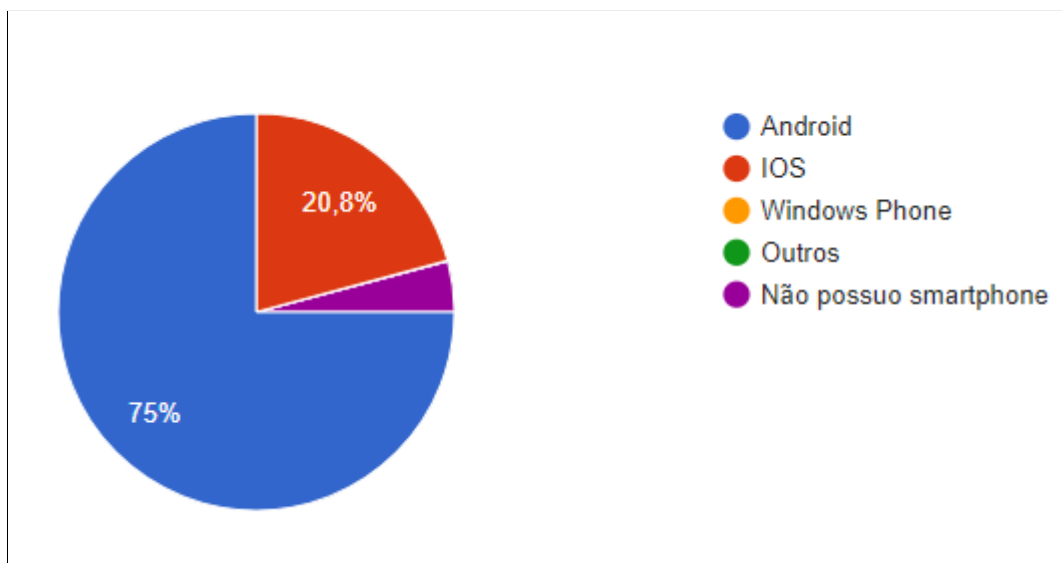


Fonte: Autor.

Ao observar a Figura 3.3 nota-se que a maioria dos participantes, cerca de 58,3%, relataram possuir o diabetes do tipo 1, enquanto que 25% deles possuíam o diabetes do tipo 2 e 16,7% eram pré-diabéticos. Nesta pesquisa não houveram casos de diabetes gestacional. O fato de que nesta pesquisa tenha maior quantidade de diabéticos do tipo 1 do que do tipo 2 pode ser explicado por ser a maioria jovem com idade menor que 20 anos. Geralmente o diabetes tipo 2 é diagnosticado próximo à quarta década de vida do paciente, embora existam casos de pessoas mais jovens com diabetes tipo 2 (OLIVEIRA, 2017).

Quando perguntados sobre com que frequência os participantes usavam o *smartphone*, 91,7% disseram que utilizavam-no várias vezes ao dia. Enquanto que uma pessoa relatou que utilizava ao menos uma vez por dia e somente uma pessoa não possui *smartphone*. Aproximadamente 95% dos entrevistados possuíam acesso a um *smartphone*. O gráfico da Figura 3.4 exhibe as plataformas móveis dos *smartphones* participantes.

Figura 3.4 – Gráfico das plataformas móveis dos smartphones dos participantes.



Fonte: Autor.

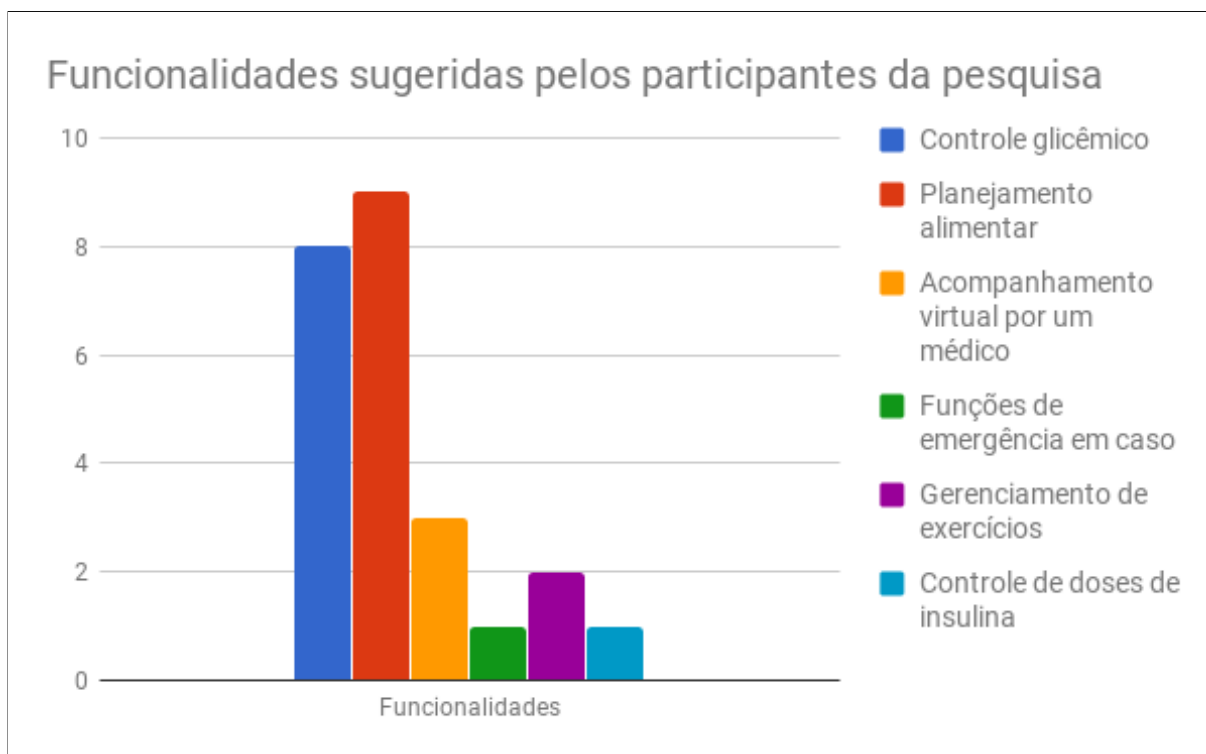
Como pode-se notar na Figura 3.4, entre todos os *smartphones* dos entrevistados, a maioria das plataformas móveis deles são Android (75%). O iOS da Apple aparece em 20% deles e apenas um dos participantes não possui *smartphone*. Deste que possuem *smartphones* 62% utilizavam algum aplicativo para auxiliar no tratamento do Diabetes. Os nomes dos aplicativos utilizados são:

- Diabetes M;
- GlicoCare (Contagem de Carboidratos);
- FatSecret (Contagem de Carboidratos);
- MySugr (Diário Glicêmico);
- Tecnonutri (Contagem de carboidratos);
- GlucoTrends;
- Glic online.

Algumas pessoas relataram utilizar mais de um aplicativo para gerenciar o diabetes. Sendo que o mais utilizado é o Diabetes M. que fora citado por três pessoas.

Conforme a Figura 3.5, dentre as funcionalidades mais importantes nos aplicativos de autocuidado do diabetes, o planejamento alimentar foi a mais indicada pelos participantes na pesquisa, logo depois, o controle glicêmico é apontado como o segundo mais importante seguido pela funcionalidade de acompanhamento por um médico.

Figura 3.5 – Gráfico das funcionalidades mais importantes para os usuários com diabetes.

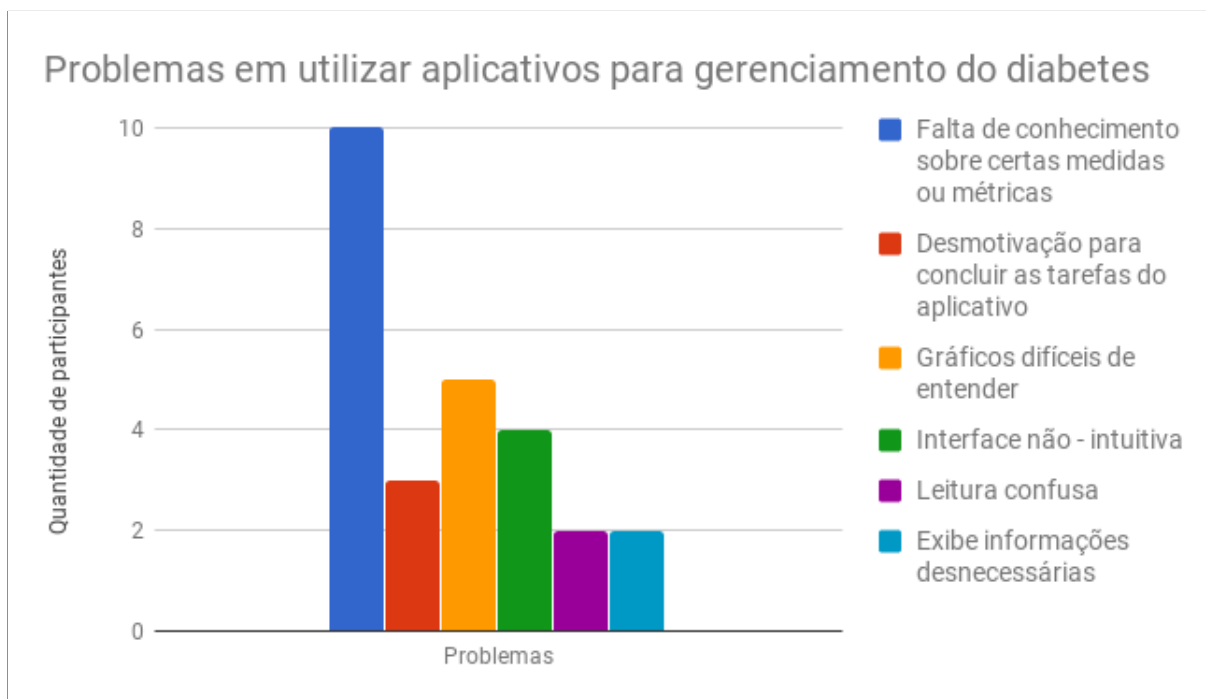


Fonte: Autor.

Embora os aplicativos de celulares sejam muito úteis no tratamento do diabetes, existem ainda problemas que impedem que o usuário tenha uma boa experiência de uso do aplicativo.

A Figura 3.6 exhibe os problemas relatados na pesquisa. A falta de conhecimento sobre métricas e medidas no tratamento da diabetes foi o maior problema descrito pelos participantes da pesquisa. O segundo fator que mais atrapalha a boa utilização de aplicativos é a presença de gráficos complexos. E em terceiro lugar vemos a falta de interfaces intuitivas, ou seja, a falta de interfaces mais fáceis de entender o funcionamento do aplicativo.

Figura 3.6 – Gráfico dos problemas encontrados em aplicativos.



Fonte: Autor.

Continuando as observações da Figura 3.6, cerca de três pessoas indicaram a desmotivação para concluir as metas impostas pelo aplicativo como um problema. Ainda duas pessoas informaram que enfrentam o problema de leitura confusa nos aplicativos, como também, exibem informações desnecessárias. Ao perguntar se os participantes da pesquisa utilizariam um aplicativo para *smartphone* que gerenciasse todas as suas informações da Diabetes e previsse o risco de comorbidades (pé diabético, problemas de visão, problemas renais, etc), cerca de 87,5% se mostraram interessados em utilizar este aplicativo. Destes 87,5%, cerca de 18,2% estariam dispostos a pagar acima de R\$ 16,00 para adquirir o aplicativo, 27% estaria disposto a pagar entre R\$ 9,00 e R\$ 10,00. Ainda 22,7% estariam dispostos a pagar menos que R\$ 9,00. E por fim, 31,8% preferem não pagar pelo aplicativo.

3.2 Considerações do capítulo

Foi visto neste capítulo os detalhes de uma pesquisa feita pela internet através de formulários web. O seu grande objetivo foi obter um olhar geral do uso de aplicações móveis focadas no cuidado do diabetes e testar a o grau de interesse em uma aplicações que buscasse solucionar os principais problemas apontados na pesquisa, como também, reunir as funcionalidades mais importantes para as pessoas com diabetes. No próximo capítulo são relatados os resultados da revisão sistemática feita para levantar o uso de aplicações móveis no autocuidado da DM.

4 Revisão sistemática

Neste capítulo são descritas a metodologia e os resultados colhidos da revisão sistemática sobre intervenções com uso de tecnologias móveis para o autocuidado pacientes com Diabetes Mellitus.

Segundo Kitchenham (2004), uma revisão sistemática tem por objetivo identificar, avaliar e interpretar todos os estudos relevantes a uma questão, fenômeno ou área de interesse em uma pesquisa. Biolchini et al. (2005) ressalta que, através da revisão sistemática, é possível coletar informações sobre o estado da arte de um determinado tema de estudo, sem sofrer com os dilemas do viés do pesquisador. O processo de revisão sistemática segue o modelo descrito na Figura 4.1.

Figura 4.1 – Processo da revisão sistemática.



Fonte: Autor.

Como mostra a Figura 4.1, a revisão sistemática foi dividida em sete etapas: definição dos objetivos, elaboração das perguntas, identificação dos termos chave, construção das strings de busca, seleção de estudos, extração de dados e apresentação dos resultados. Este processo é linear, ou seja, cada etapa é dependente do resultado da etapa anterior.

4.1 Metodologia da Revisão Sistemática

Com base em Kitchenham (2004) e Biolchini et al. (2005), foi criada uma revisão sistemática com o objetivo de identificar intervenções envolvendo o uso de aplicativos móveis no cuidado do Diabetes *Mellitus*, com o propósito de reunir informações as funcionalidades oferecidas pelas aplicações, ações de tratamento envolvidas e resultados. Dentro deste escopo, foram criadas as seguintes questões de pesquisa:

$\mu 0$: Quais são as técnicas de tratamento do diabetes que estão sendo aplicadas por meio de aplicações móveis?

$\mu 1$: Quais funcionalidades são oferecidas pelas aplicações móveis para o cuidado da Diabetes descritos na literatura?

O quadro 4.1 mostra os termos chave extraídos do objetivo e das questões de pesquisa.

Quadro 4.1 – Palavras chaves para construção das strings de busca.

Tipo de estudo	Aplicações móveis	Autocuidado	Diabetes
<i>intervention</i>	<i>mobile application, mobile phones, cell phone-based software</i>	<i>treatment, self care, self-management, attention, health</i>	<i>diabetes</i>

Fonte: Autor.

A partir dos termos chave descritos no Quadro 4.1 foram criadas strings de busca para serem utilizadas em bases de dados digitais. A revisão foi feita nas seguintes bases de dados:

- IEEE Xplore¹;
- PubMed Central: PMC²;
- Medline Complete (Ebsco)³;
- ScienceDirect ⁴;
- Scopus⁵.

¹ <http://ieeexplore.ieee.org/Xplore/home.jsp>

² <https://www.ncbi.nlm.nih.gov/pmc/>

³ <https://health.ebsco.com/products/medline-complete>

⁴ <https://www.sciencedirect.com/>

⁵ <https://www.scopus.com/home.uri>

As strings de busca estão dispostas na Tabela 4.1 e estão ao lado das suas respectivas bases.

Tabela 4.1 – Strings de busca.

Base Digital	String de busca
String genérica	("intervention") AND ("mobile application"OR "mobile phones"OR "cell phone-based software") AND ("treatment"OR "self care"OR "self-management"OR "attention"OR "health") AND ("diabetes")
IEEE Xplore	(.QT.intervention.QT.) AND (.QT.mobile application.QT. OR .QT.mobile phones.QT. OR .QT.cell phone-based software.QT.) AND (.QT.treatment.QT. OR .QT.self care.QT. OR .QT.self-management.QT. OR .QT.attention.QT. OR .QT.health.QT.) AND (.QT.diabetes.QT.)
PubMed Central: PMC	"(intervention"[AB]) AND ("mobile application"[AB] OR "mobile phones"[AB]) AND ("treatment"[AB] OR "self care"[AB] OR "self-management"[AB] OR "attention"[AB] OR "health"[AB]) AND "diabetes"[AB] AND ("2013/04/08"[PubDate] : "2018/04/06"[PubDate]) AND ("2013/04/08"[PDat] : "2018/04/06"[PDat])
Medline Complete (Ebsco)	("intervention") AND ("mobile application"OR "mobile phones"OR "cell phone-based software") AND ("treatment"OR "self care"OR "self-management"OR "attention"OR "health") AND ("diabetes") Data de publicação: 20130101-20181231; Língua inglesa"
ScienceDirect (Elsevier)	("intervention") AND ("mobile application"OR "mobile phones"OR "cell phone-based software") AND ("treatment"OR "self care"OR "self-management"OR "attention"OR "health") AND ("diabetes")
Scopus	TITLE-ABS-KEY (("intervention") AND ("mobile application"OR "mobile phones"OR "cell phone-based software") AND ("treatment"OR "self care"OR "self-management"OR "attention"OR "health") AND ("diabetes")) AND (LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013)) AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (SUBJAREA , "MEDI")) AND (LIMIT-TO (LANGUAGE , "English"))

Fonte: Autor.

Na execução da primeira busca de artigos, os estudos foram selecionados com base nos parâmetros passados na *string* de busca. Os estudos retornados pela primeira busca, foram submetidos à etapa de inclusão. Nesta etapa foram escolhidos estudos primários que satisfizessem os seguinte critérios:

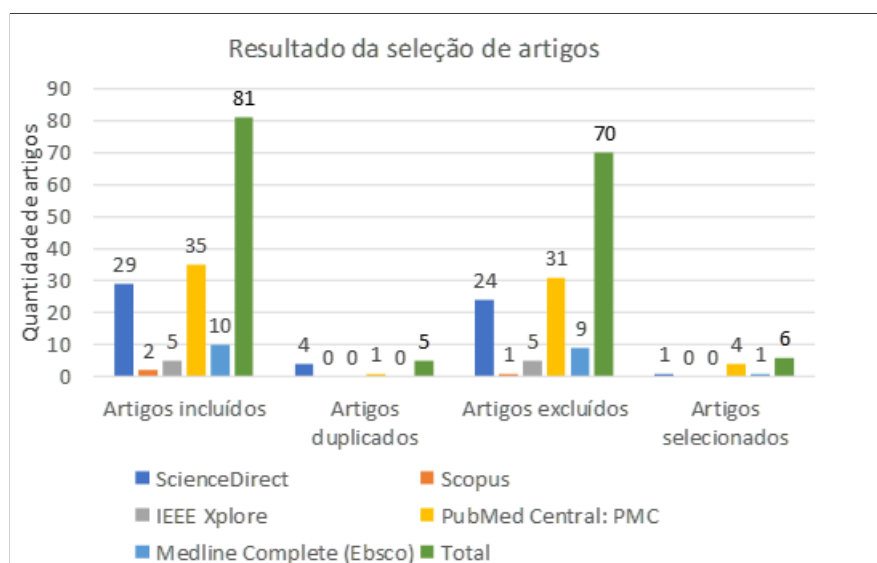
- Serão incluídos os artigos que foram publicados entre 2013 e 2018 (6 anos até a data da busca);
- Somente artigos de *journals* e revistas acadêmicas serão incluídos;
- Os artigos devem conter em seus abstracts ou palavras-chaves os termos da string de busca.

Cerca de 81 artigos passaram pela etapa de inclusão e avançaram para a etapa de exclusão. Os artigos que passaram pela etapa de exclusão foram avaliados segundo os seguintes critérios:

- Os artigos devem estar escritos em inglês;
- Os artigos duplicados serão excluídos;
- Os artigos devem estar disponíveis para download;
- Os artigos devem contemplar, em seu texto, ao menos um estudo de caso de intervenção de uma aplicação móvel ou mobile focada no cuidado da Diabetes.

Os artigos que não foram condizentes com ao menos um dos critérios de exclusão foram retirados da revisão. Os detalhes estão dispostos na Figura 4.2.

Figura 4.2 – Resultados das etapas de inclusão e exclusão.



Fonte: Autor.

A Figura 4.2 mostra um gráfico com as quantidades de artigos por base, dentro de cada etapa de seleção dos artigos. Ao final da etapa de exclusão, foram removidos 70 artigos, restando 6 para a etapa de extração de dados, e assim, responder as questões de pesquisa. O Quadro 4.2 exhibe os artigos selecionados para a revisão.

Quadro 4.2 – Artigos selecionados.

Referência	Título	Autor(res)
Alotaibi, Istepanian e Philip (2016)	A mobile diabetes management and educational system for type-2 diabetics in Saudi Arabia (SAED)	Alotaibi, Mohammed M Istepanian, Robert Philip, Nada.
Desveaux et al. (2016)	A randomized wait-list control trial to evaluate the impact of a mobile application to improve self-management of individuals with type 2 diabetes: a study protocol	Desveaux, Laura Agarwal, <i>et al</i>
Maglalang e Yoo (2017)	Acceptability and Cultural Relevance of a Mobile Health Lifestyle Intervention for Filipinos with Type 2 Diabetes	Maglalang, Dale Dagar Yoo, Grace.
Nundy et al. (2013)	Developing a behavioral model for mobile phone-based diabetes interventions	Nundy, Shantanu Dick, Jonathan J. Solomon, Marla C. Peek, Monica.
Poppe et al. (2017)	Users' thoughts and opinions about a self-regulation-based eHealth intervention targeting physical activity and the intake of fruit and vegetables: A qualitative study	Poppe, Louise Van der Mispel, Celien De Bourdeaudhuij, Ilse Verloigne, Maïté Shadid, Samyah Crombez, Geert.
Stuckey et al. (2013)	A lifestyle intervention supported by mobile health technologies to improve the cardiometabolic risk profile of individuals at risk for cardiovascular disease and type 2 diabetes: study rationale and protocol	Stuckey, Melanie I Shapiro, Sheree Gill, Dawn P Petrella, Robert J.

Fonte: Autor.

Para responder a questão de pesquisa $\mu 0$ foram reunidas todas as estratégias de tratamento que foram apoiadas por aplicações móveis. O Quadro 4.3 exhibe quais foram estas técnicas relacionadas a cada estudo.

Quadro 4.3 – Técnicas de tratamento do Diabetes aplicadas na intervenção.

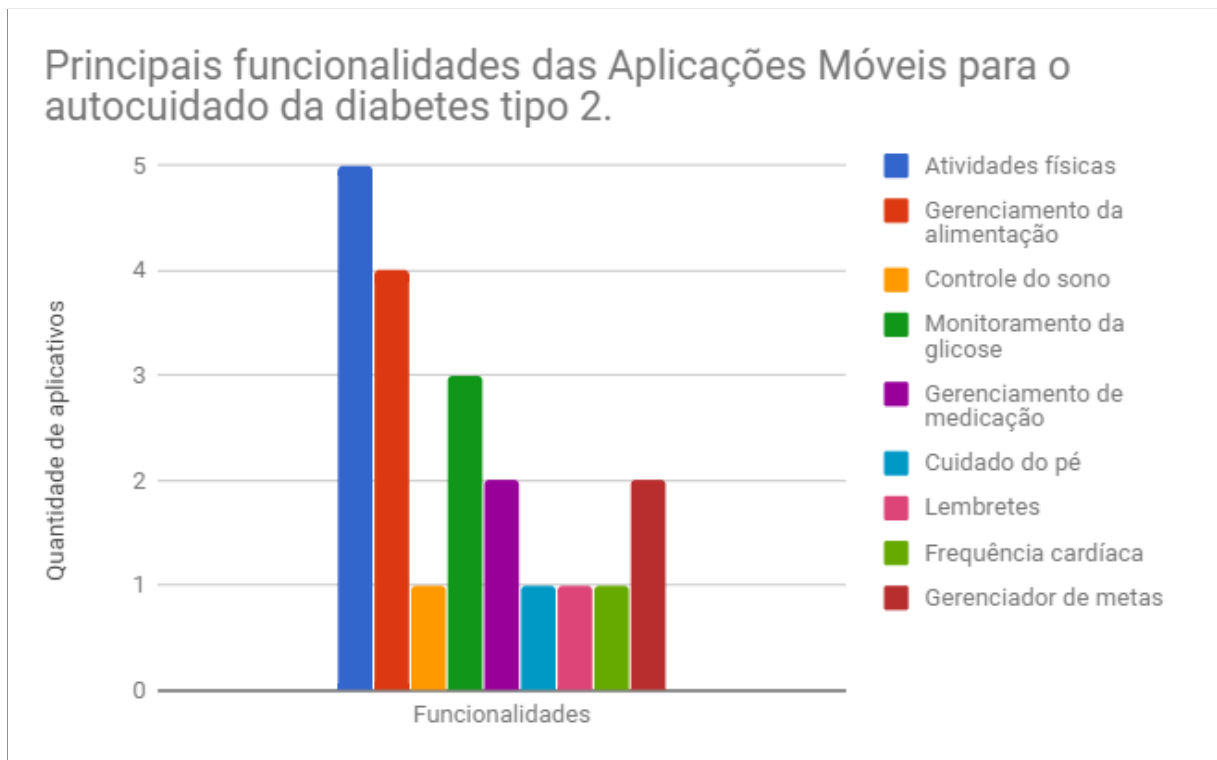
Referências	Técnicas de tratamento do DM2
Alotaibi, Istepanian e Philip (2016)	Registros de nível de glicose no sangue. Gerenciamento de Lembretes. Mensagens relacionadas a temas do tratamento da diabetes (Tratamento, monitoramento de glicose no sangue, diabetes, complicações, melhoria do atendimento clínico e dieta). Comunicação com um médico por meio de mensagens.
Desveaux et al. (2016)	Educação sobre os níveis de glicose no sangue. Monitoramento de complicações. Nutrição. Atividades físicas. Estratégias de proteção vascular e otimização do controle glicêmico.
Maglalang e Yoo (2017)	Redução dos riscos de síndrome metabólica através da prática regular de exercícios físicos.
Nundy et al. (2013)	Cuidado do pé. Monitoramento da glicose.
Poppe et al. (2017)	Uso de atividades físicas. Ingestão de frutas e legumes.
Stuckey et al. (2013)	Exercícios físicos com acompanhamento médico.

Fonte: Autor.

No Quadro 4.3, além do tratamento comum ligado ao controle glicêmico (exercícios regulares, boa alimentação, uso de medicamentos), é possível notar o uso de intervenções com aplicativos móveis para tarefas de cuidado com o pé (NUNDY et al., 2013) e monitoramento de complicações do diabetes (DESVEAUX et al., 2016), que são os grandes problemas encontrados pela maioria dos pacientes com nível de glicemia não controlado.

Para responder a segunda questão de pesquisa, foram reunidas as funcionalidades apresentadas pelas aplicações móveis. O gráfico da Figura 4.3 mostra as funcionalidades identificadas nos aplicativos.

Figura 4.3 – Artigos selecionados por ano de publicação.



Fonte: Autor.

O gráfico da Figura 4.3 mostra que a funcionalidade que é mais presente nas aplicações alvo das intervenções é o gerenciamento de atividades físicas, logo após vem o gerenciamento da alimentação e em terceiro lugar se encontra o monitoramento da glicose. É possível notar que algumas funcionalidades para cuidado de complicações como cuidado do pé é observado em uma das aplicações.

4.2 Resumos dos artigos selecionados

Nesta seção são descritos os detalhes de cada intervenção, abordando os seus objetivos, metodologias aplicadas e os resultados encontrados.

4.2.1 *Developing a behavioral model for mobile phone-based diabetes interventions*

Nundy et al. (2013) abordam um estudo com o objetivo de Explorar os mecanismos potenciais pelos quais uma intervenção baseada no telefone celular melhorou o auto-gerenciamento do diabetes e desenvolver um modelo comportamental que pode ser usado para projetar e avaliar aplicativos móveis de saúde.

Foram realizadas 18 entrevistas em profundidade, individuais, semi-estruturadas em diversos pontos de vista entre os afro-americanos com Diabetes. Participaram da pesquisa cerca de 18 pessoas. Elas forneceram dados a um programa de autogestão de diabetes baseado em mensagem de texto, durante 4 semanas. Cada entrevista durou aproximadamente 60 min. Foram selecionados pacientes adultos com Diabetes que usam hipoglicemiantes orais e/ou insulina, que possuísem telefone celular e que não estivessem hospitalizados até 2 meses antes da pesquisa. Os participantes receberam US \$ 25 para cobrir os custos de mensagens de texto e US \$ 30 para participação. As informações sociodemográficas e clínicas foram coletadas na inscrição, assim como os dados de linha de base sobre o uso de telefone celular e mensagens de texto.

Como resultado, apenas uma pequena maioria estava confortável com mensagens de texto no início do estudo (56%). Os participantes descreveram tornar-se mais conscientes do seu diabetes e mais no controle de sua doença. Eles também refletiram sobre estarem mais conscientes das mudanças fisiológicas em sua condição. As mensagens reforçaram o sucesso dos participantes em cuidar de seu diabetes, seja de *feedback* explícito de uma mensagem (por exemplo, "excelente trabalho") ou indiretamente por meio de evidências de que as mensagens de texto estavam melhorando sua saúde. Os participantes reconheceram uma maior conscientização de seus risco pessoal de complicações e da relação entre atividades de autocuidado e desfechos de diabetes. O programa também ajudou os participantes a aceitar sua condição e entender o significado de ter uma doença crônica. Alguns participantes acreditavam que as mensagens de texto estavam sendo enviadas diretamente do administrador de texto, e não de um programa de computador automatizado, enquanto outras conscientemente antropomorfizavam o programa de mensagens de texto.

4.2.2 A lifestyle intervention supported by mobile health technologies to improve the cardiometabolic risk profile of individuals at risk for cardiovascular disease and type 2 diabetes: study rationale and protocol

Stuckey et al. (2013) testaram os efeitos de uma intervenção de exercício prescritiva, contra uma intervenção apoiada pela tecnologia de saúde móvel em fatores de risco cardiometabólico. Este estudo foi realizado por meio de um ensaio clínico randomizado controlado com dois braços de intervenção, um grupo recebeu prescrição de exercícios e um kit de tecnologias móveis de saúde, enquanto que o outro recebeu apenas prescrição de exercícios adaptados. Participaram da pesquisa 149 pessoas, adultos com idade entre 18 e 70 anos que apresentava dois riscos fatores de risco de síndrome metabólica confirmado por testes clínicos.

Os programas intensivos de intervenção no estilo de vida reduziram risco de desenvolver Diabetes mellitus tipo 2 em pacientes com pré-diabetes e melhorias foram mantidas durante um longo período de acompanhamento. O exercício físico de resistência reduziu a circunferência da cintura, a pressão arterial sistólica e a pressão arterial diastólica e aumentou o colesterol

lipoproteico de alta densidade, sem alterações na glicose ou triglicérides plasmáticos em jejum.

4.2.3 *A mobile diabetes management and educational system for type- 2 diabetics in Saudi Arabia (SAED)*

Alotaibi, Istepanian e Philip (2016) Testaram o nível de eficácia de um sistema inteligente de gerenciamento de diabetes móvel e adaptado para pacientes com DM2 na Arábia Saudita, projetado com o objetivo de melhorar o autogerenciamento da doença.

O estudo piloto foi conduzido durante o período de 6 meses na região de Tabuk na Arábia Saudita. O teste clínico foi conduzido com um total de 20 participantes. Os pacientes foram separados aleatoriamente dentro de dois grupos, um Grupo Intervenção (GI) e um Grupo Controle(GC), com 10 pacientes cada. Os grupos foram constituídos por pacientes portadores de diabetes tipo 2 com idade entre 20 e 65 anos. O GI utilizou o sistema SAED para coletar dados do nível de glicose no sangue 2 a 3 vezes ao dia, antes e depois das refeições, em 2 ou 3 vezes na semana. O teste HbA1c foi feito no início e no fim do período de 6 meses. O GC recebeu tratamento de forma tradicional.

Ao analisar os resultados, pode se observar que houve uma redução significativa nos resultados de HbA1c no final do período de 6 meses para o grupo de intervenção SAED, quando comparado com o grupo controle. A média inicial de HbA1c para o grupo de intervenção foi de 8,76% (tolerância de 0,76%) e 8,61% para o grupo controle. A linha de base média do grupo de intervenção diminuiu para 7,85% de 8,76% no final do período de 6 meses. É importante notar que todos os pacientes do grupo de intervenção mostraram melhorias muito boas no conhecimento sobre diabetes, enquanto no grupo de controle há grandes inconsistências com cada paciente.

4.2.4 *A randomized wait-list control trial to evaluate the impact of a mobile application to improve self-management of individuals with diabetes: a study protocol*

Desveaux et al. (2016) realizaram uma intervenção com objetivo avaliar se uma aplicação mobile projetada para melhorar o auto-gerenciamento, pode contribuir para a melhoria de resultados clínicos comparados ao cuidado usual em pacientes com Diabetes *Melitus*.

Foi projetado um modelo de teste de controle em lista de espera, no qual dois grupos de pessoas foram separados para execução da intervenção. Um grupo receberá a intervenção com a aplicação móvel de imediato, e o outro receberá tratamento por cuidado tradicional nos 3 primeiros meses, depois, receberá a intervenção com o aplicativo móvel. Este teste é feito durante o período de 6 meses.

Para fazer parte dos grupos de pesquisa, o paciente precisa obedecer os seguintes critérios: ser adulto acima dos 18 anos de idade, estar em tratamento da Diabetes *Mellitus* ou estar participando em um Programa de Educação da Diabetes, HbA1c $\geq 8.0\%$ (e ao menos 1% acima do nível dos participantes da pesquisa) no resultado clínico mais recente entre os últimos 3 meses, possuir uma conta de *email* ou estar apto a obter um e possuir a habilidade de ler em inglês. Foram excluídos da amostra os paciente que estiveram em monitoramento contínuo da glicose, possuíam uma bomba de insulina, estavam em diálise, gestantes ou que fosse incapaz de usar um computador ou telefone celular devido a grave deficiência mental ou física. Os dados foram colhidos através de questionário aplicado no início e fim dos 3 primeiros meses e no fim dos 6 meses.

4.2.5 *I don't have to explain, people understand*": acceptability and cultural relevance of a mobile health lifestyle intervention for filipinos with type 2 diabetes

Maglalang e Yoo (2017) avaliaram a aceitabilidade e relevância cultural do programa PilAm Go4Health - uma intervenção de estilo de vida de saúde móvel adaptada culturalmente, incluindo redes sociais virtuais e apps para filipinos americanos com Diabetes.

Foram selecionadas 45 pessoas para participar da intervenção com aplicativos móveis durante um período de 6 meses. Elas foram solicitadas a usar uma pulseira acelerômetro *Fitbit* diariamente, relatar a ingestão de alimentos / calorias e peso usando o aplicativo *Fitbit* diário e participar de grupos privados no Facebook. Os participantes foram escolhidos a partir dos seguintes critérios de seleção: se declarar Filipino, possuir massa corporal $> 23 \text{ kg/m}^2$, ser homem ou mulher com idade > 18 anos, ser diagnosticado com Diabetes (confirmado por dados clínicos: glicemia em jejum $> 126 \text{ mg / dL}$ ou um Teste Oral de Tolerância à Glicose positivo - TOTG positivo), não ser dependente de tratamento com insulina, sem comprometimento cognitivo baseado no teste Mini-Cog, nenhuma deficiência física impedindo a caminhada de 20 minutos continuamente, falar em inglês, possuir um smartphone, não ter participado no programa de modificação do estilo de vida no último ano, não possuir doença associada ao metabolismo da glicose (por exemplo, síndrome de Cushing, Acromegalia e Feocromocitoma em tratamento), não ter sido tratado recentemente por causa de doenças na tireoide, não possuir distúrbio alimentar conhecido, não apresentar condições médicas conhecidas, como infarto do miocárdio, requerendo programa de exercícios especiais, e não possuir membro da família inscrito no estudo. Os dados foram colhidos ao final da intervenção por meio de um questionário com 10 perguntas.

Todos os participantes sentiram que a tecnologia promoveu uma nova forma de assistente pessoal para o auto-gerenciamento da diabetes. Todos os participantes sentiram uma necessidade profunda de mudar seus estilos de vida para evitar complicações que afligem seus entes queridos. Ao contrário das tentativas anteriores de mudança, eles descobriram que o aplicativo *Fitbit*

forneceu informações instantâneas e educação para facilitar mudanças com hábitos alimentares e níveis de atividade física. Para 64,4% dos entrevistados essa intervenção forneceu uma nova perspectiva sobre o manejo da diabetes.

4.2.6 Users' thoughts and opinions about a self-regulation-based eHealth intervention targeting physical activity and the intake of fruit and vegetables: A qualitative study

Poppe et al. (2017) investigam as perspectivas dos usuários em um programa baseado no web site "MyPlan 1.0", que visa aumentar a prática de atividades físicas e a ingestão de frutas e vegetais.

Foram selecionadas 40 pessoas que foram divididas igualmente em dois grupos, um grupo contendo pessoas com Diabetes e outro com pessoas sem diabetes. Em cada grupo existia a mesma quantidade de homens e mulheres, com idades acima dos 18 anos, que falam Holandês. Para participar do grupo com pacientes de Diabetes, precisava comprovar que tinha a doença por meio de teste clínico lançado a mais de um mês. Os dados foram coletados através de formulário qualitativa.

No grupo de pessoas com Diabetes, cerca de 65% concordaram que ter um estilo de vida saudável tem um efeito positivo em seu bem-estar físico e mental, enquanto outros discordam. Ainda, 75% afirmou que a criação de planos de ação era desnecessária. Além disso, os usuários muitas vezes acreditavam que a criação de planos específicos é difícil devido a alterações nos horários das semanas. Quanto a usabilidade, os participantes relataram dificuldades para preencher os questionários para avaliar as atividades físicas, ingestão de frutas ou legumes. Todos os participantes acharam difícil saber como responder as perguntas e duvidaram que estivessem realmente fazendo o que o site solicitava que fizessem.

4.3 Considerações do capítulo

Este capítulo mostrou a execução de cada etapa do processo da revisão sistemática. Esta revisão teve como objetivo levantar evidências sobre intervenções com aplicativos móveis para melhorar o tratamento de pessoas com Diabetes. Ao analisar os resultados pôde-se observar a presença de técnicas de tratamento voltadas para o autocuidado de algumas complicações como cuidado com o pé através de uma aplicativo móvel. O próximo capítulo descreve a metodologia utilizada e os resultados obtidos no estudo das aplicações semelhantes.

5 Estudo de Aplicações Semelhantes

Este capítulo aborda a pesquisa de soluções de aplicativos semelhantes focadas no Autocuidado do Diabetes, que teve por objetivo elicitare as principais funcionalidades encontradas em aplicações muito populares. Esta Revisão de Mercado foi feita na *Google Play*¹, loja de aplicativos oficial da *Google*, como o trabalho foi focado primeiramente no desenvolvimento na plataforma Android, desconsideramos a busca por aplicações na *Apple Store*.

5.1 Metodologia de Pesquisa

Dentro da loja *Google Play* existem 4 grandes categorias de produtos que são oferecidos como Filmes, Musicas, Livros e Aplicativos, por isso, foi selecionado o produto Aplicativo. O filtro Categoria remete à área ao qual o aplicativo esteja relacionado, desta forma, optou-se por buscar aplicações relacionadas a área de medicina. A String de busca é a frase utilizada no buscador da *Google Play*. O filtro Tipo de monetização foi incluído pois a aplicação tema deste projeto se enquadra como uma aplicação gratuita, o que facilita o acesso dos usuários. A *Google Play* possui uma métrica de avaliação através de estrelas, que variam de 1 a 5, afim de encontrar aplicativos bem avaliados, optou-se por incluir na pesquisa somente aplicativos que possuíssem no mínimo 4 estrelas. O último critério de seleção é a quantidade de downloads, não é uma métrica muito precisa, mas subtemde o seu nível de popularidade, por isso, foram selecionadas aplicações com no mínimo 10.000 downloads. Por fim, as aplicações selecionadas satisfizeram os seguintes critérios apresentados no Quadro 5.1.

Quadro 5.1 – Filtros utilizados na Revisão de Mercado.

Filtro	Critério
Tipo de produto.	Aplicativo
Categoria.	Medicina.
String de busca.	App para diabéticos
Tipo de monetização.	Gratuito.
Nível de avaliação.	Acima de 4 estrelas.
Quantidade de downloads	Mais de 10.000 downloads.

Fonte: Autor.

No Quadro 5.2, ao lado esquerdo, são apresentadas as aplicações selecionadas para o estudo de aplicações semelhantes, ao lado direito, estão dispostos os seus desenvolvedores.

¹ <https://play.google.com/store/apps>

Quadro 5.2 – Descrição dos Aplicativos Aceitos na Revisão de Mercado.

Titulo	Desenvolvedor
mySugr: Diário da Diabetes	mySugr GmbH
Diabetes:M	Sirma Medical Systems
SocialDiabetes. Take control of your Diabetes	SocialDiabetes

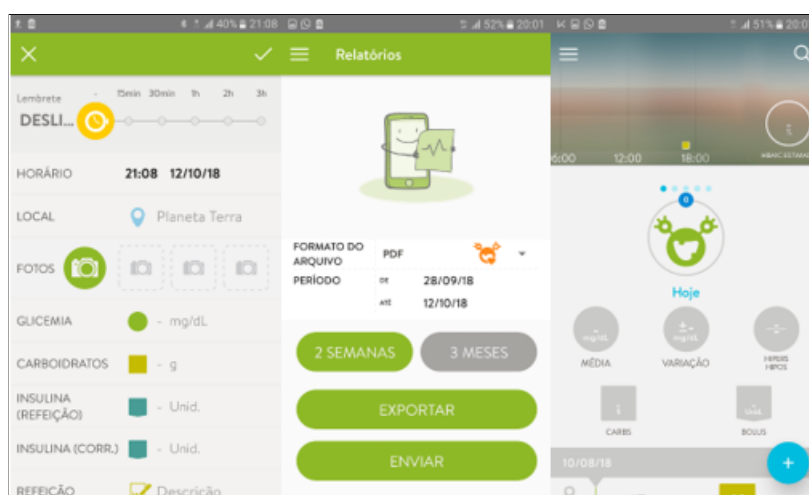
Fonte: Autor.

As aplicações presentes no Quadro 5.2 são detalhadas nas próximas seções quanto as suas funcionalidades, ao final do capítulo estas funcionalidades são agrupadas e apresentadas em um quadro.

5.2 mySugr: Diário da Diabetes

O *mySugr: Diário da Diabetes* é um aplicativo que auxilia o paciente diabético a ter um melhor monitoramento da sua doença. Dentre suas funções ele é capaz de guardar informações que são inseridos pelos usuários como os níveis de glicemia, ingestão de carboidratos, unidades de insulinas, entre outras. O aplicativo possui uma interface intuitiva tornando fácil o aprendizado do usuário quanto ao uso do aplicativo. O aplicativo é exibido na Figura 5.2.

Figura 5.1 – Aplicativo mySugr: Diário da Diabetes.



Fonte: Adaptado de (MYSUGR GMBH, 2018).

Ele contempla ainda funções de lembretes e emissão de relatórios, possibilitando ao usuário analisar e guardar suas medições, juntamente com um profissional de saúde (caso possível).

O *mySugr*: Diário da Diabetes traz uma funcionalidade diferencial dos demais aplicativos, que é a rede de estatísticas que possibilita ao usuário verificar níveis glicêmicos em semana, quinzenas, mês e trimestre. Vale constar que as principais funções não necessitam de conexão com a internet. A Figura 5.2 exibe a funcionalidade de estatísticas glicêmicas.

Figura 5.2 – Aplicativo mySugr: Diário da Diabetes.

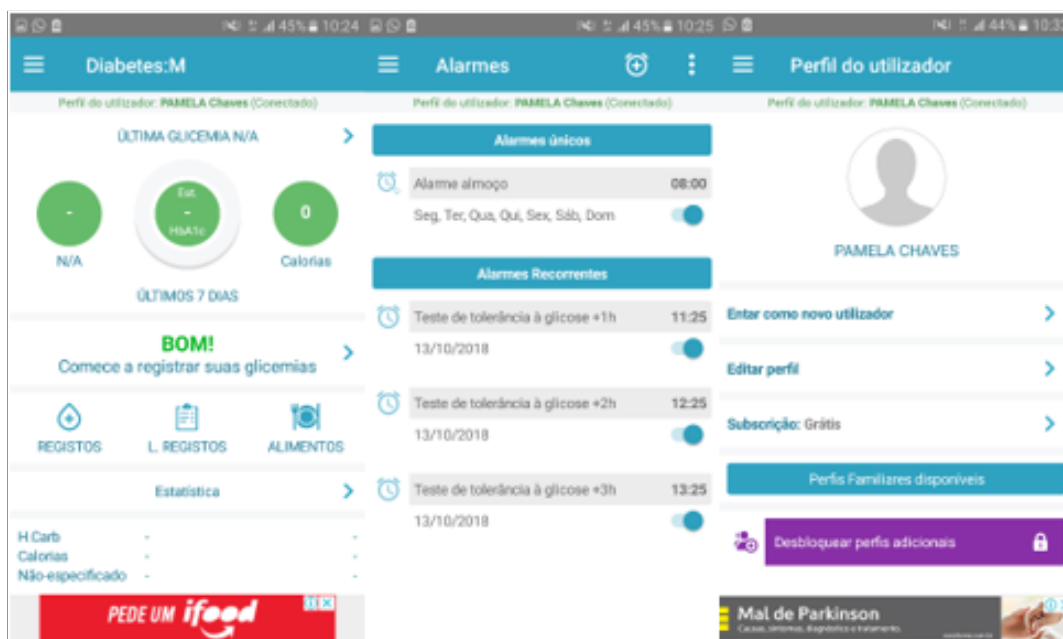


Fonte: Adaptado de (MYSUGR GMBH, 2018).

5.3 Diabetes:M

O *Diabetes:M* é um aplicativo para pacientes diabéticos, que facilita o acompanhamento da doença. O usuário pode usufruir de funções como registros dos níveis de glicemia, alimentação, uso das insulinas e medicamentos orais. O aplicativo possui uma interface simples e de fácil aprendizagem. Ele possui ainda funções de guardar relatórios e alarmes que auxiliam na vida cotidiana dos pacientes/usuários. A Figura 5.3 exibe algumas destas funcionalidades.

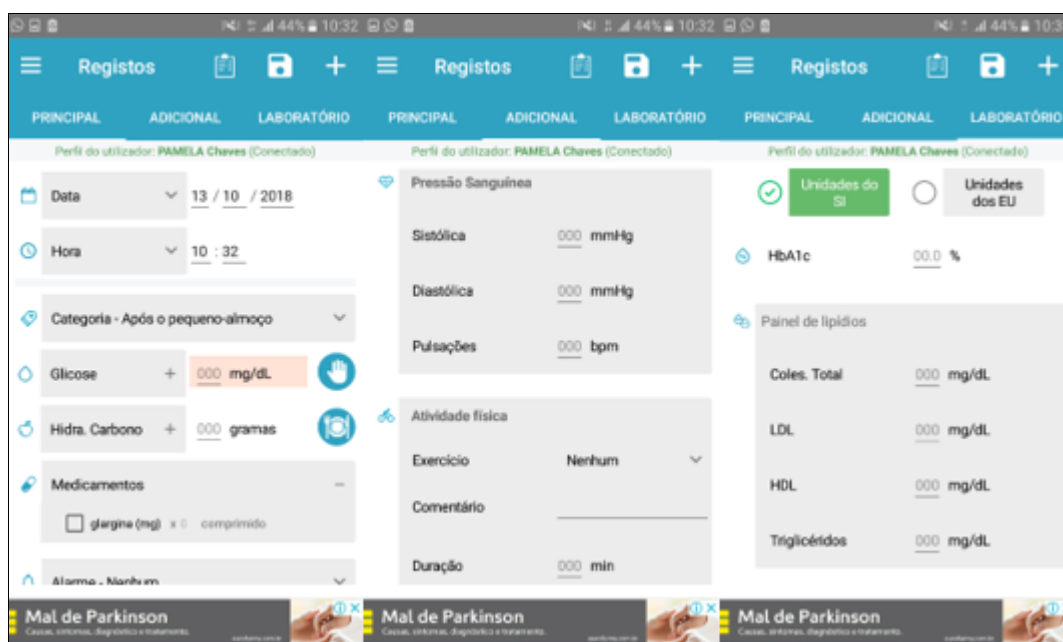
Figura 5.3 – Aplicativo Diabetes:M.



Fonte: Adaptado de (SIRMA MEDICAL SUSTEMS, 2018).

É importante ressaltar que para os usuários possam utilizar mais funções do aplicativo, o mesmo deve adquirir a versão *Premium*. Vale ressaltar que as principais funções não necessitam de internet. A Figura 5.4 mostra as funcionalidades de registro da Glicose, Pressão Sanguínea, exercícios e Níveis de Hemoglobina Glicada.

Figura 5.4 – Demais funcionalidades do Aplicativo Diabetes:M.



Fonte: Adaptado de (SIRMA MEDICAL SUSTEMS, 2018).

5.4 SocialDiabetes

SocialDiabetes. Take control of your Diabetes é um aplicativo que ajuda no monitoramento do paciente diabético. Várias funções estão contidas nessa ferramenta, entre elas: registro dos níveis de glicemia, alimentação, tratamento por insulina ou medicamentos orais, registro de exercícios físicos, entre outras. Ele possui uma interface dinâmica com a exibição configurável dos componentes. Contempla ainda as médias glicêmicas por dia, semana e mês. Além disso, pontos de hipoglicemias do mês passado em relação ao mês atual e uma estimativa da Hemoglobina Glicada (HbA1c). A Figura 5.5 exibe estas funcionalidades.

Figura 5.5 – Aplicativo *SocialDiabetes*.



Fonte: Adaptado de (SOCIAL DIABETES, 2018).

5.5 Considerações sobre os aplicativos semelhantes

A busca por soluções semelhantes proporcionou a compreensão das funcionalidades mais comuns dos aplicativos focados no gerenciamento do Diabetes. Este proporcionou um entendimento de quais funcionalidades o aplicativo deve focar e como ele deve se diferenciar dos outros. O Quadro 5.3 agrupa as principais características de cada aplicativo.

Quadro 5.3 – Características extraídas dos aplicativos da Revisão de Mercado.

-	mySugr: Diário da Diabetes	Diabetes:M	SocialDiabetes. Take control of your Diabetes
Gerenciamento de Nível de Glicemia	X	X	X
Ingestão de Carboidratos	X	X	X
Gerenciamento de Insulina	X	X	X
Gerenciamento de Medicamentos Orais		X	X
Exportar Relatórios	X		
Registros de Exercícios Físicos		X	X
Sistema de Lembretes/Alarmes	X	X	X
Hemoglobina Glicada Estimada			X

Fonte: Autor.

Como mostra o Quadro 5.3 as funcionalidades mais comuns entre os aplicativos selecionados foram o Gerenciamento de Nível de Glicemia, Ingestão de Carboidratos, Gerenciamento de insulina e o Sistema de Lembretes/Alarmes. Somente o *mySugr* não possui alguma funcionalidade de gerenciamento de medicamentos Orais. O *mySugar* apresentou uma funcionalidade de exportação de dados, mas não oferece o armazenamento de registros de Exercícios Físicos. Quanto a Estimativa da Hemoglobina Glicada, apenas a *SocialDiabetes* possui esta funcionalidade.

5.6 Considerações do capítulo

Neste capítulo foi mostrado os resultados da Revisão de Mercado que mostrou as aplicações semelhantes ao TiaBete. Três aplicações passaram pela etapa de seleção que foram: *mySugr: Diário da Diabetes*, *Diabetes:M* e *SocialDiabetes. Take control of your Diabetes*. Foram extraídas informações destas aplicações que, ao final do capítulo, gerou um quadro com as principais funcionalidades encontradas entre elas. No próximo capítulo serão abordadas as etapas do desenvolvimento que foram necessárias para a construção do aplicativo, descrevendo os artefatos resultantes de cada fase do *RUP*.

6 Desenvolvimento do aplicativo

Neste capítulo é descrito todo o processo de desenvolvimento do aplicativo Tia Bete, utilizando os conceitos do processo de desenvolvimento *RUP*. Primeiro é dada uma visão geral do projeto, descrevendo o seu escopo, requisitos funcionais e não funcionais e principais *stakeholders*. Este projeto foi dividido em nove iterações, onde cada iteração possui o seu ciclo de fases baseado no *RUP*. O processo de desenvolvimento foi agrupado em quatro fases para cada iteração: Concepção, Elaboração, Construção e Transição. Cada uma destas fases agrupa um conjunto de artefatos do sistema que descrevem os casos de uso, estruturas, comportamentos, dados, questões de segurança e comunicação presentes no aplicativo. Este capítulo descreve o desenvolvimento da 1ª interação do protótipo de software, segmentando-o nas fases do *RUP*.

6.1 Visão Geral do Produto e Projeto

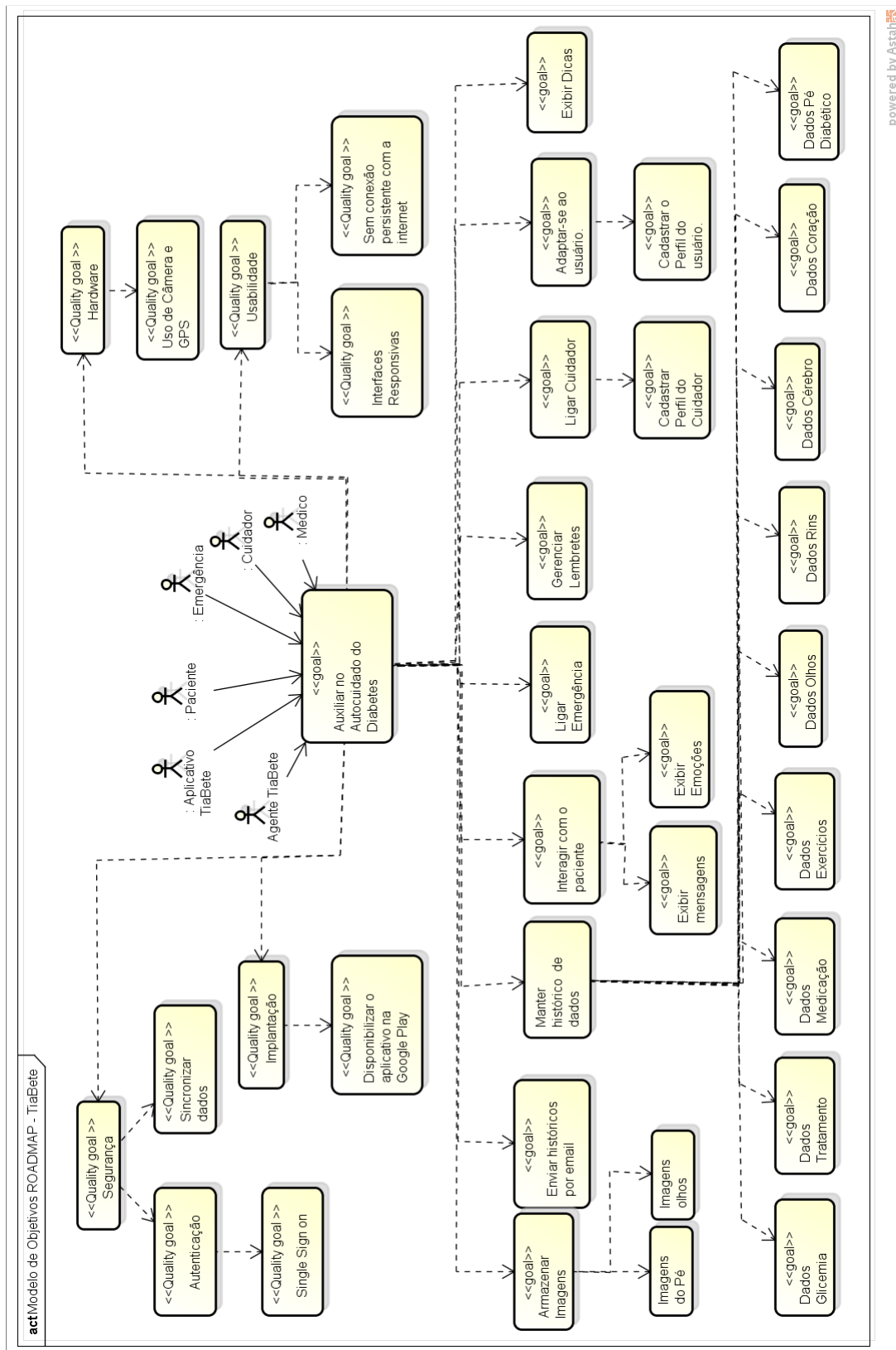
Como foi demonstrado no Capítulo 2, cerca de 425 milhões de pessoas foram diagnosticadas com diabetes em 2017, sendo que no Brasil, aproximadamente 73,2% dos diabéticos não efetuam o correto controle do Diabetes. A melhor forma de diminuir os índices de mortalidade da doença é aumentando a educação do paciente quanto ao seu autocuidado do diabetes.

Para isso, foi construído um protótipo de uma assistente virtual para *smartphones Android* que auxilie nas tarefas de autocuidado do diabetes, semelhante a um cuidador virtual, onde este reúne as principais informações do usuário e traz dicas de como melhorar o autocuidado e que auxilia na tomada de decisão quanto a mudança do comportamento. O aplicativo foi projetado para pessoas entre as idades de mais de 45 anos de idade, onde segundo a (INTERNATIONAL DIABETES FEDERATION, 2017), é a faixa onde mais se predomina a identificação da diabetes.

Segundo Sommerville (2011), a primeira atividade do processo de engenharia de software a ser executada é a Engenharia de Requisitos. É através dela que são descritas as funcionalidades que o sistema deverá fazer, os serviços que ele fornecerá, como também as restrições do próprio sistema. Ao final, é gerado um conjunto de requisitos de software que é dividido em duas categorias: Requisitos Funcionais (RF) e Requisitos Não-Funcionais (RNF).

Para Cheah et al. (2017), a Engenharia de Software de sistemas orientados a agentes consistem de vários passos que devem ser apoiados por um modelo de objetivos, modelo de papéis e um modelo de organização que devem ser analisados para a construção de aplicações móveis. Em nosso problema não foi necessário o uso de modelos de organização e modelos de papéis, uma vez que, o aplicativo não mostrou relações complexas entre os papéis e não se insere em um ambiente organizacional. Desta forma foi elaborado o Modelo de *ROADMAP* que é exibido na Figura 6.1.

Figura 6.1 – Modelo de Objetivos ROADMAP.



Fonte: Autor.

A Figura 6.1 exibe os papéis do projeto, os objetivos de alto nível, subobjetivos e objetivos de qualidade que são necessários para apoiar os objetivos. Todos os objetivos são gerados a partir do objetivo principal "Auxiliar no Autocuidado do Diabetes", que exibe todos os papéis relacionados ao aplicativo, os seus objetivos de qualidade e subobjetivos da aplicação. Este Modelo *ROADMAP* foi gerado a com base nos componentes do Diagrama de Atividades da UML. Os símbolos do modelo *ROADMAP* são representados por meio de esteriótipos sobre as atividades, os relacionamentos são representados por setas de dependência da UML.

Os papéis, que são representados na Figura 6.1 por "pessoas palito", que podem ser utilizados para determinar os atores da aplicação. O projeto da aplicação foi construído com base nestes seis papéis, onde o papel "Aplicativo TiaBete" representa um sistema de software que compartilha das informações do Paciente ao "Agente TiaBete", que representa a Assistente Virtual. Estes papéis são detalhados mais a frente na seção 6.1.3.

Para solucionar os problemas quanto à deficiência do autocuidado em pacientes com Diabetes, foram necessários criar diversos subobjetivos. Tomando como exemplo o objetivo "Interagir com o Paciente" do *ROADMAP* da Figura 6.1, para que o sistema pudesse atingir o objetivo "Auxiliar no autocuidado do Diabetes", dentre todos os subobjetivos, foi necessário atingir o objetivo "Interagir com o paciente", mas para isso, é necessário que o sistema satisfaça o subobjetivo "Exibir mensagens", como também o subobjetivo "Exibir emoções", que remetem as animações que expressam alguma emoção da assistente virtual.

Os objetivos de qualidade refletem os Requisitos Não-Funcionais da aplicação, como por exemplo, para atender objetivo de qualidade "Segurança", é necessário atingir os objetivos de qualidade "Autenticação" e "Sincronizar dados", ambos são detalhados na etapa de Elicitação de Requisitos Não - Funcionais 6.1.2.

Os objetivos "Enviar histórico por email", "Manter históricos dos pacientes", "Gerenciar Lembretes" e "Exibir Dicas" e "Manter histórico de dados" são objetivos comuns a todos os aspectos envolvidos no tratamento do Diabetes (Nível Glicêmico, Alimentação, Medicação, Exercícios, Olhos, Coração, Cérebro, Rins e Pé diabético). Desta forma, foi necessário criar modelos de i* Objetivos sobre a perspectiva de cada um destes aspectos.

Na Figura 6.2 é demonstrado o modelo de i* Objetivos para Nível Glicêmico, mais especificamente, em relação à Glicemia Capilar. Ele foi criado em cima dos componentes utilizados no Diagrama de Atividades da UML. Os componentes receberam esteriótipos que condizem o seu respectivo símbolo do Modelo i* Objetivo proposto por Cheah et al. (2017). Os relacionamentos de dependência do modelo de i* Objetivos são apresentados pelo relacionamento de dependência da própria UML, cujo conceito permanece o mesmo. O círculo envolta do ator determina os seus limites, como por exemplo, o ator Agente TiaBete, detém a responsabilidade de alcançar os objetivos "Manipular Mensagens", "Manipular animações", "Interagir com o paciente", "Adaptar-se ao usuário", "Processar dados" e "Acessar Meta dados".

O modelo de *i** Objetivos da Figura 6.2 descreve a organização dos relacionamentos entre os atores, objetivos e recursos, através de uma visão de mais baixo nível, útil para validação dos desenvolvedores.

Os papéis vistos na Figura 6.1 foram mapeados para *actors*, que são representações de pessoas ou sistemas. Neste modelo os *actors* agrupam objetivos e recursos que são necessários para delimitar o seu ambiente de responsabilidades. Os *actors* importantes para o projeto são: Paciente, Medico, Emergência, Cuidador, Agente TiaBete e o Aplicativo Tiabete. O Paciente, para enviar um histórico de dados de glicemia através do Aplicativo TiaBete, o aplicativo precisa manter um histórico dos dados glicêmicos do Paciente, como por exemplo, os dados da Glicemia Capilar, que por sua vez, necessita do cadastro dos registros de aferições da Glicemia Capilar.

Os objetivos ou *goals*, possuem a mesmas definições dos objetivos presentes no *ROADMAP*. Por exemplo, para que o Paciente possa ver as mensagens e animações que o Agente TiaBete exibe, primeiro, o agente precisa alcançar o objetivo "Interagir com o paciente", que por sua vez, necessita da conclusão dos subobjetivos "Exibir mensagens" e "Adaptar-se ao usuário" e "Processar dados" e assim segue-se as relações de dependências.

Os recursos representam componentes pré-existentes necessários para alcançar objetivos (CHEAH et al., 2017). Na Figura 6.2 os recursos "Gerenciar Lembretes", "Utilizar Telefone" e "Gerenciar Contatos" são recursos oferecidos, neste caso, pelo ambiente da aplicação que é a plataforma *Android*.

Após a construção do modelo de objetivos do modelo *ROADMAP* e do modelo de *i** objetivos, foram desempenhadas as atividades de Modelagem de Negócio e de Engenharia de Requisitos, resultando na delimitação do escopo, identificação dos Atores, construção do modelo *Use - Case* da aplicação e na elicitação dos Requisitos Funcionais e Não-Funcionais.

6.1.1 Requisitos Funcionais

Sommerville (2011) fala que os Requisitos Funcionais representam as funcionalidades que o sistema deve fornecer ao seus usuários. Estas funcionalidades devem levar em consideração o negócio, os seus envolvidos (*stakeholders*) e também as tecnologias que serão utilizadas. Existem dois tipos de Requisitos Funcionais propostos por Sommerville (2011): os Requisitos de Usuário e os Requisitos de Sistema. Os Requisitos de Usuário (RU) são requisitos de alto nível, que compreendem a visão dos *stakeholders*, já os Requisitos de Sistema (RS), abordam a visão dos profissionais engajados na construção do sistema. O Quadro 6.1 mostra os RUs elicitados durante as reuniões presenciais com os *stakeholders* do projeto:

Quadro 6.1 – Requisitos Funcionais do aplicativo.

Código	Nome	Descrição	Prioridade
RF001	Manter nível de glicose	Armazenar informações sobre o nível de glicose do paciente (hemoglobina glicada, glicemia normal em jejum).	Essencial
RF002	Manter medicação	Registrar horário, nome do medicamento e dose administrada.	Essencial
RF003	Manter atividades físicas	Armazenar duração das atividades e descrição da atividade.	Essencial
RF004	Manter alimentação	Armazenar informações sobre as refeições, como horário, calorias consumidas e frequência.	Essencial
RF005	Mostrar Dicas	Falar frases para ajudar o usuário a entender como está o seu tratamento que será dito pela assistente virtual.	Essencial
RF006	Manipular Emoções da Tia Bete	Executar a mudança de comportamento da assistente virtual a depender do envolvimento do usuário.	Essencial
RF007	Manter informações sobre os olhos	Manter dados do paciente sobre o estado da visão e fotos da retina.	Prioritário
RF008	Manter informações sobre o coração	Manter dados sobre data e horário, pressão arterial antes e após eventos, e batimentos cardíacos.	Prioritário
RF009	Manter informações sobre nervos	Registrar acontecimentos como desmaios, sonolência e dor no corpo.	Prioritário
RF010	Manter informações sobre o pé	Armazenar dados sobre o pé do paciente como fotos, aparecimento de feridas.	Prioritário
RF011	Manter informações sobre os rins	Mantém o registro de quantas vezes o paciente urinou durante o dia, estado da urina, frequência em que bebe água.	Prioritário
RF012	Manter notificações	Armazenar as notificações não atendidas.	Prioritário
RF013	Chamar emergência	Oferecer funcionalidades para ligar para números de emergência.	Desejável
RF014	Chamar cuidador	Ligar e mandar mensagens para um contato automaticamente para pedir socorro.	Desejável

Fonte: Autor.

O Quadro 6.1 é o resultado da última iteração da fase de Concepção. Para isto, foram analisados os resultados das pesquisas de campo do Capítulo 3, um Doutor em Bioquímica pela UFS e um discente em Farmácia pela UFS.

6.1.2 Requisitos Não-Funcionais

Pressman e Maxim (2016) definem Requisitos Não-Funcionais ou RNFs como atributos de qualidade, de desempenho, de segurança ou como uma restrição geral de um sistema. Estes requisitos aumentam a qualidade do software e o torna utilizável para o usuário final. O Quadro 6.2 exibe todos os RNFs identificados.

Quadro 6.2 – Requisitos Não-Funcionais do aplicativo.

Código	Nome	Descrição	Prioridade	Tipo
RNF001	Criar animações 2D.	Projetar animações gráficas em 2D da Tia Bete e projetar também os estados de "alegria" e "frustração".	Essencial	Usabilidade
RNF002	Uso de telas responsivas.	A interface deve ser capazes de se adaptar ao tamanho da tela e rotação.	Essencial	Usabilidade
RNF003	Sem conexão persistente com a internet	A aplicação deve oferecer todas as suas funcionalidades sem necessitar o acesso à internet.	Prioritário	Usabilidade
RNF004	Os dados serão sincronizados com um servidor de dados na nuvem.	A aplicação deve ser capaz de sincronizar as informações com um servidor na nuvem a fim de criar um backup dos dados.	Prioritário	Segurança
RNF005	Login <i>single sign on</i>	Criar um sistema de login simplificado com <i>Sing Sign On</i> .	Prioritário	Segurança
RNF006	Diponibilizar para versões superiores a 5.0/5.1 (Lolipop).	A aplicação será mobile para o sistema Android a partir da versão 5.0/5.1 (Lolipop).	Prioritário	Implantação
RNF008	Publicar o aplicativo na <i>Google Play</i> .	Publicar uma versão alfa e publicá-la na loja virtual <i>Google Play</i> .	Desejável	Implantação
RNF009	Uso de câmeras e <i>GPS</i> .	O aplicativo deve ser desenvolvido apenas para <i>smartphones</i> que suportem o uso de câmeras e <i>GPS</i> .	Desejável	Hardware

Fonte: Autor.

Após identificados os RNFs do Quadro 6.2, foi criado o projeto dos casos de uso do sistema. As definições dos casos de uso e de sua modelagem são descritos no próximo tópico que consta a fase de Modelagem do Negócio.

6.1.3 Modelagem de Negócio

Nesta atividade foram identificados os principais Atores e os casos de uso do aplicativo em uma visão de alto nível, capaz de ser compreendida pelos *stackholders*. Segundo (SOMMERVILLE, 2011), os Atores podem ser qualquer tipo de pessoa ou sistemas, que são representados por uma figura "palito". Eles são utilizados para mostrar onde ocorrerá cada interação dos usuários com as funcionalidades e qual o tipo de usuário que irá interagir com ela. Esta relação é feita ligando uma linha reta entre o ator e o balão da funcionalidade. O Quadro 6.3 mostra os principais Atores identificados durante a fase de modelagem de Negócio.

Quadro 6.3 – Principais Atores.

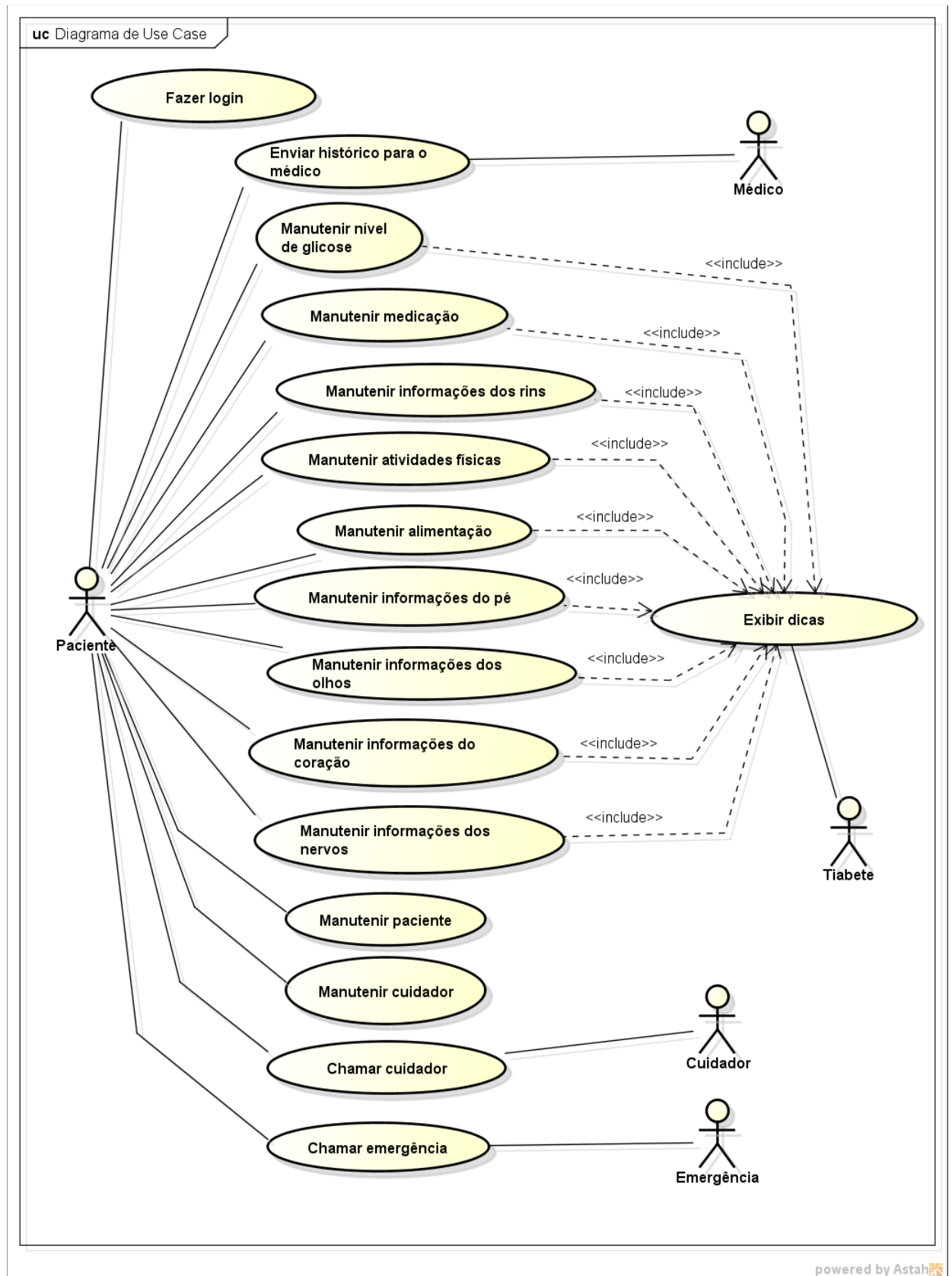
Atores	Descrição
Paciente	Pessoa diagnosticada com Diabetes <i>Mellitus</i> com idade maior que 40 anos.
Emergência	Central de atendimento responsável por fornecer o serviço de emergência.
Cuidador	Pessoa íntima do paciente que pode ser um familiar ou profissional de saúde e que está pronto para ajudá-lo em casos de emergência.
Médico	Profissional de saúde responsável por acompanhar o tratamento do paciente.
TiaBete	Agente de software responsável por avaliar informações do tratamento do paciente.

Fonte: Autor.

Embora o ator Paciente represente uma pessoa com idade acima de 40 anos, isto não quer dizer que não existam pessoas com diabetes abaixo desta faixa etária. Este Ator apenas mostra o principal perfil de usuário do aplicativo. O ator Emergência não representa um usuário em específico, mas sim, um provedor de serviço ao qual tem uma função passiva na aplicação, pois, ele apenas é contactado pelo paciente. O Médico de forma semelhante ao ator de Emergência, atua de forma passiva, ou seja, ele apenas recebe os relatórios via e-mail caso o paciente os queira enviar.

Estes Atores apresentados no Quadro 6.3, estão vinculados aos principais casos de uso do aplicativo que são representados pelo Diagrama de *Use Case* na Figura 6.3.

Figura 6.3 – Diagrama de Use Case da aplicação da Tia Bete.



Fonte: Autor.

Ao fim do processo de modelagem de negócio, foram definidos os casos de uso do aplicativo presentes na Figura 6.3. Devido a grande quantidade e complexidade dos casos de uso em relação ao tempo e quantidade de desenvolvedores no projeto, o projeto foi segmentado em nove iterações com o objetivo de poder criar ao final da iteração um protótipo. O Quadro 6.4 mostra os casos de uso a serem desenvolvidos em cada iteração.

Quadro 6.4 – Iterações e seus respectivos casos de uso.

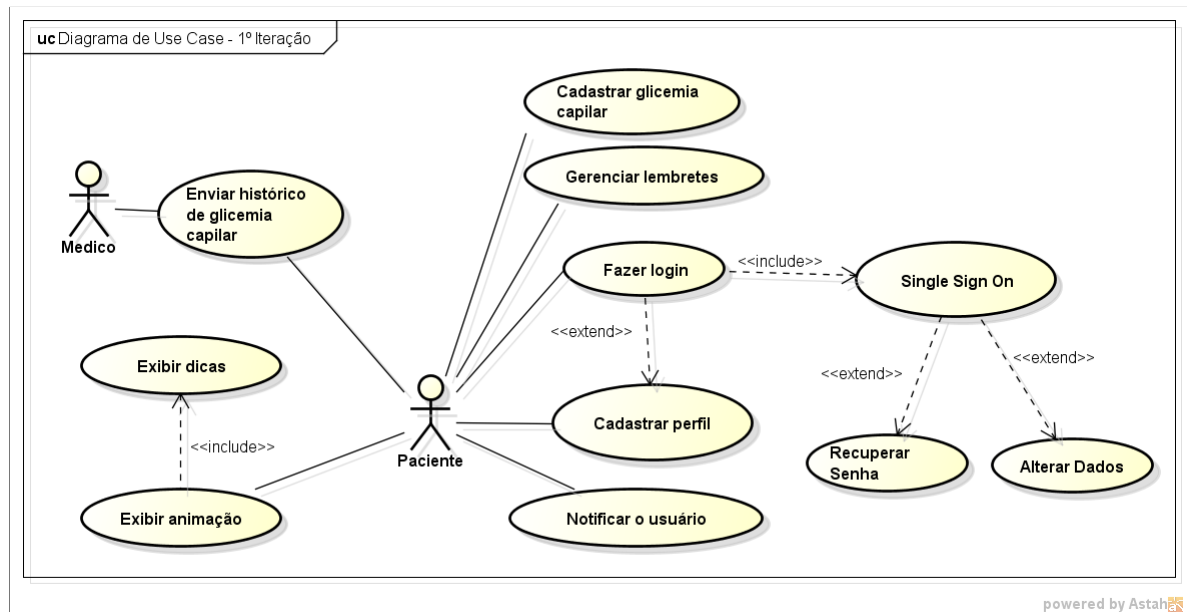
Iterações	Casos de uso
1º Iteração	Fazer login Manutenir nível de glicose Exibir dicas Manutenir paciente Exibir expressões
2º Iteração	Manutenir alimentação
3º Iteração	Manutenir medicação
4º Iteração	Manutenir exercícios
5º Iteração	Manutenir informações dos olhos Manutenir informações do pé
6º Iteração	Manutenir informações do coração
7º Iteração	Manutenir informações dos nervos
8º Iteração	Manutenir informações dos rins
9º Iteração	Manutenir cuidador Chamar cuidador Chamar emergência

Fonte: Autor.

Este trabalho focou em mostrar a execução da 1º Iteração, pois, ela demonstra o fluxo completo do aplicativo e prepara a arquitetura para os outros casos de uso, de forma a diminuir o tempo de desenvolvimento deles.

6.2 Fase de Concepção da 1º Iteração

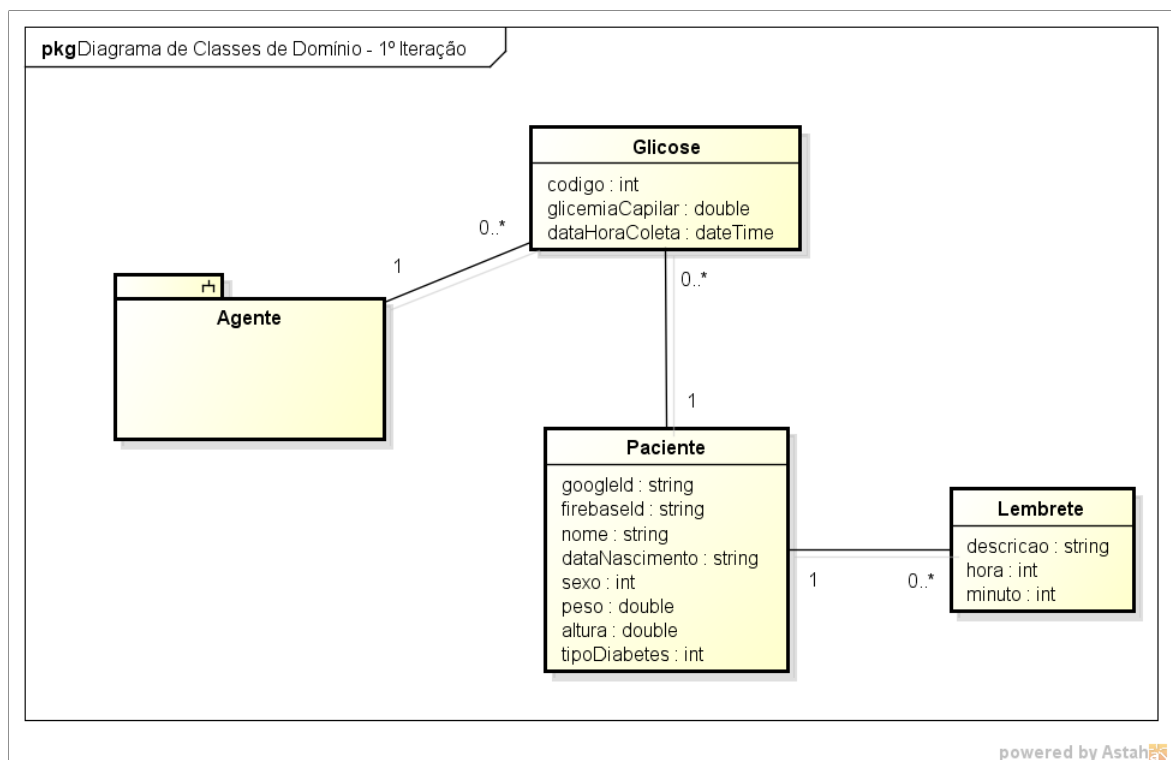
Tomando por base os casos de uso da 1º Iteração visto no Quadro 6.4, foi projetado um Diagrama de *Use Case* detalhado para facilitar o entendimento dos desenvolvedores. A Figura 6.4 demonstra esse diagrama detalhando os casos de uso da Iteração 1.

Figura 6.4 – Diagrama de *Use Case* da 1ª Iteração.

Fonte: Autor.

A partir do Diagrama de *Use Case* da Figura 6.4 foi criado um Diagrama de Classes de Domínio que mostra as relações entre os principais componentes conceituais do sistema. A Figura 6.5 descreve o Diagrama de Classes de Domínio para a 1ª iteração.

Figura 6.5 – Diagrama de Classes de Domínio da 1ª Iteração.

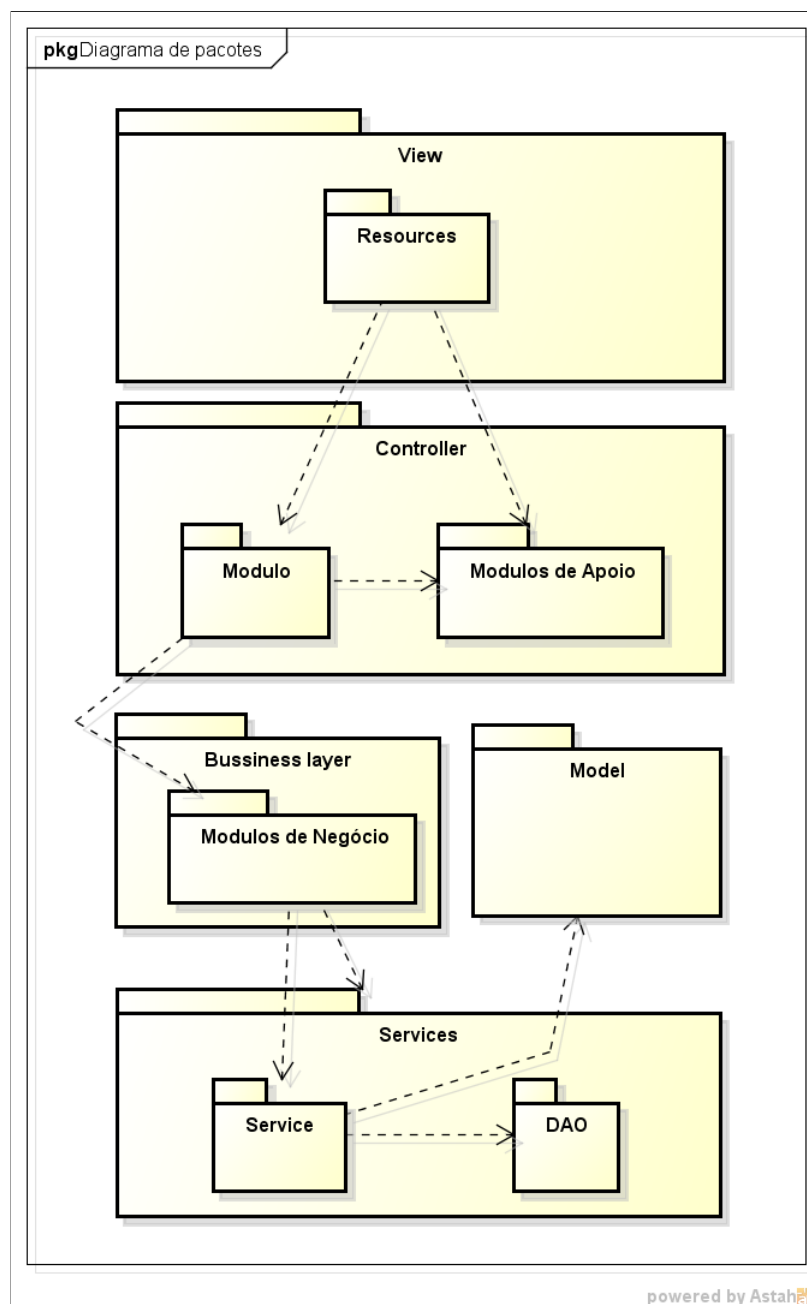


Fonte: Autor.

6.3 Fase de Elaboração da 1ª Iteração

A arquitetura de software foi criada a partir de uma derivação do *framework* MVC (*Model - View - Controller*), adicionando duas camadas ao *framework*: a *Bussiness Layer* e a *Service*. A *Bussiness Layer* é a camada que reúne código voltado as regras de negócio e que pode ser reaproveitado entre várias plataformas móveis. A camada *Service* reúne os componentes responsáveis pela comunicação de dados, lá estão os módulos de comunicação com o banco local e serviços *web*. A arquitetura de software é demonstrada na Figura .

Figura 6.6 – Arquitetura de software.



powered by Astah

Fonte: Autor.

A arquitetura de software da Figura 6.6 foi pensada para ser flexível e adaptável a novos módulos de apoio ou de negócio, bem como a novas regras e módulos de comunicação. A partir dos artefatos construídos na fase de Elaboração, foram gerados os códigos fontes da aplicação na fase de Construção. A próxima seção descreve o que foi construído na fase de Construção.

6.4 Fase de Construção da 1ª Iteração

Etapa que antecede a fase de Transição, a Construção, é o momento no projeto onde são implementados todos os artefatos, funcionalidades e conceitos descobertos nas fases anteriores (JIN; LIANG, 2016). Neste capítulo é apresentado o código gerado através do modelo estático de Diagrama de Classes de Projeto. O Banco de dados Não-Relacional é representado por meio de um modelo de árvore. O funcionamento do código é representado por meio do Diagrama de Sequência e Diagrama de Máquina de Estados.

6.4.1 Ferramentas e Especificações Técnicas

O projeto foi implementado em código C, utilizando a IDE *Visual Studio*¹ junto a plataforma de desenvolvimento de aplicações Móveis *Xamarin*. O *Visual Studio* é um ambiente de desenvolvimento integrado criado pela *Microsoft* para o desenvolvimento de software com base em *.NET Framework* e às linguagens *Visual Basic (VB)*, *C*, *C++*, *C (C Sharp)* e *F (F Sharp)*. Também disponibiliza o desenvolvimento web por meio do *ASP.NET*, como web sites, aplicativos web, serviços web e também aplicações móveis, como é o caso do *Xamarin*.

O *Visual Studio* oferece uma versão gratuita que é chamada de *Community*, onde é possível baixá-la do próprio site da *Microsoft*. Para instalá-lo basta visitar o site² e baixar o arquivo executável. Depois de instalado é preciso preparar o ambiente quanto ao tipo de aplicação que será gerado. No *Visual Studio installer* existem vários pacotes de desenvolvimento que podem ser acoplados a IDE do *Visual Studio* para o desenvolvimento, um deles é o Desenvolvimento com *Xamarin*, que foi o utilizado neste trabalho.

A IDE se utiliza de *IntelliSense* para fazer o *auto-complete* de praticamente tudo no editor de código. Isto traz uma maior produtividade, como também, melhora a visibilidade do código e performance graças a funcionalidade de melhorias de código que o próprio *IntelliSense* oferece.

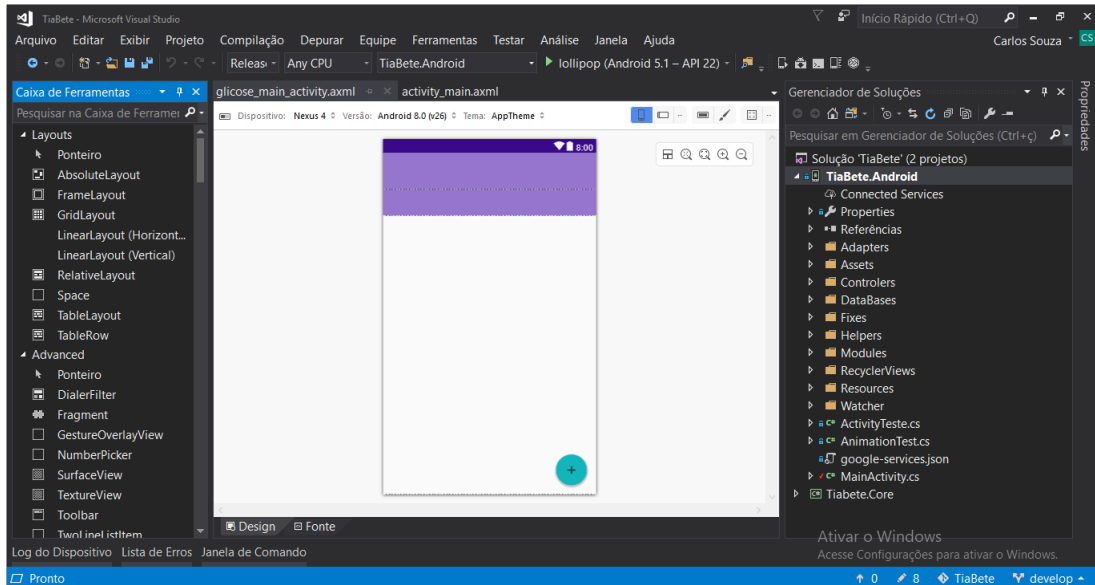
Graças ao fácil acoplamento das SDKs ao *Visual Studio*, ele consegue obter o mesmo comportamento de construção de aplicações Android como se estivesse programando no próprio *Android Studio*. Ele dispõe de uma funcionalidade de *Drag-in-Drop* (Arrastar e Soltar) para os

¹ <https://visualstudio.microsoft.com/pt-br/vs/features/ide/>

² <https://visualstudio.microsoft.com/pt-br/vs/community/>

componentes de tela de um aplicativo, semelhante ao *Drag-in-Drop* do *Android Studio*, onde os componentes que são arrastados para o compartimento do dispositivo são convertidos em código .xml, que corresponde ao código .xml do Android. A Figura 6.7 exibe a visão de um projeto Android aberto no *Visual Studio*.

Figura 6.7 – Visão de um projeto Android no *Visual Studio*.



Fonte: Autor.

Para o desenvolvimento deste aplicativo foi utilizado um notebook *Acer Aspire ES1-533*. O Quadro mostra as configurações do equipamento utilizado

Quadro 6.5 – Iterações e seus respectivos casos de uso.

Produto	Marca	Especificação
Processador	Intel	Intel(R) Celeron(R) CPU N3450 1.10GHz
Disco Rígido	Acer	500 GB WDC WD5000LPCX-21VHAT0
Memória RAM	—	4 GB (1 x 4 GB) tipo DDR3 Frequência 1600 MHz
Sistema Operacional	Microsoft	Microsoft Windows 10 Home Single Language
Sistema Operacional	Microsoft	Microsoft Windows 10 Home Single Language
IDE	Microsoft	Microsoft Visual Studio Community 2017 Versão 15.7.4

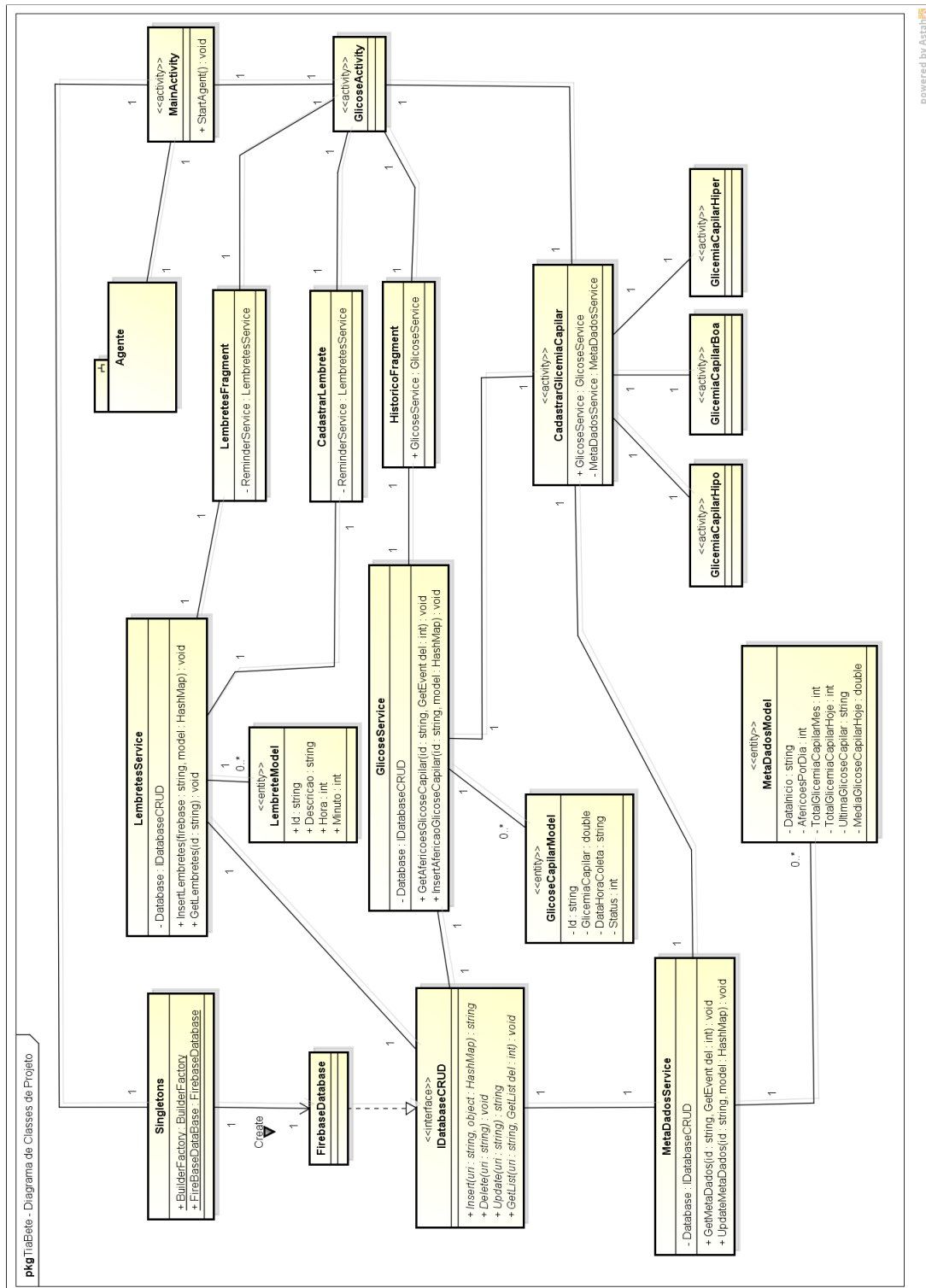
Fonte: Autor.

Através das especificações técnicas e equipamentos (Figura 6.5), foi construído o aplicativo *Android*. As próximas seções detalham o processo de construção de código.

6.4.2 Construção do Código

O código construído é representado pelo Diagrama de Classes de Projeto exibido na Figura 6.8. Nela são mostradas as estruturas implementadas com as suas relações estáticas.

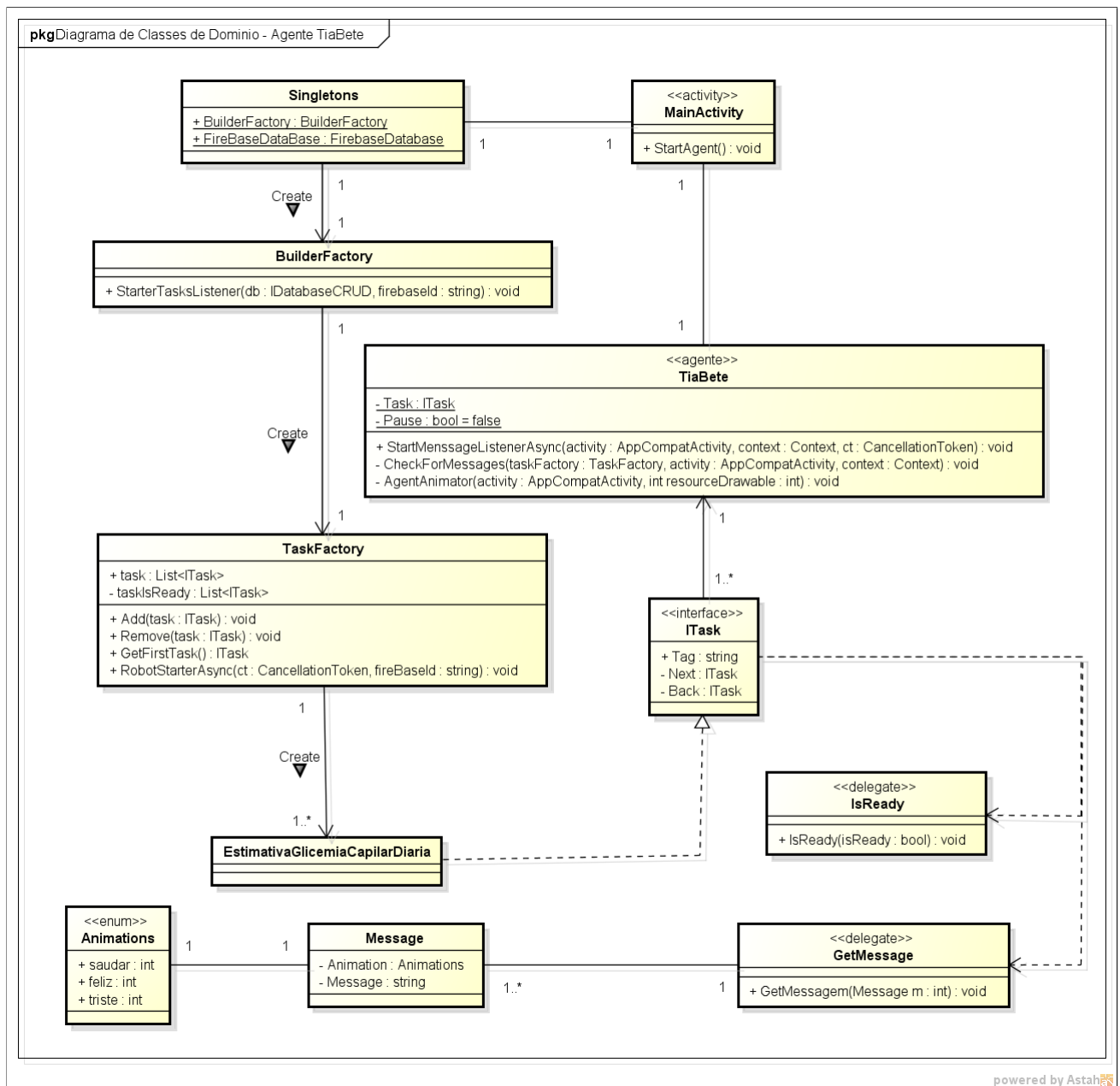
Figura 6.8 – Diagrama de Classes de Projeto da 1ª iteração.



Fonte: Autor.

Devido a natureza do projeto em ser centrado em um agente, foi necessário criar um Diagrama de Projeto para a construção do Agente TiaBete, segundo o modelo utilizado por (MARCO; PACE, 2013). A Figura 6.9 demonstra a implementação do agente TiaBete.

Figura 6.9 – Diagrama de Classes de Projeto do Agente TiaBete.

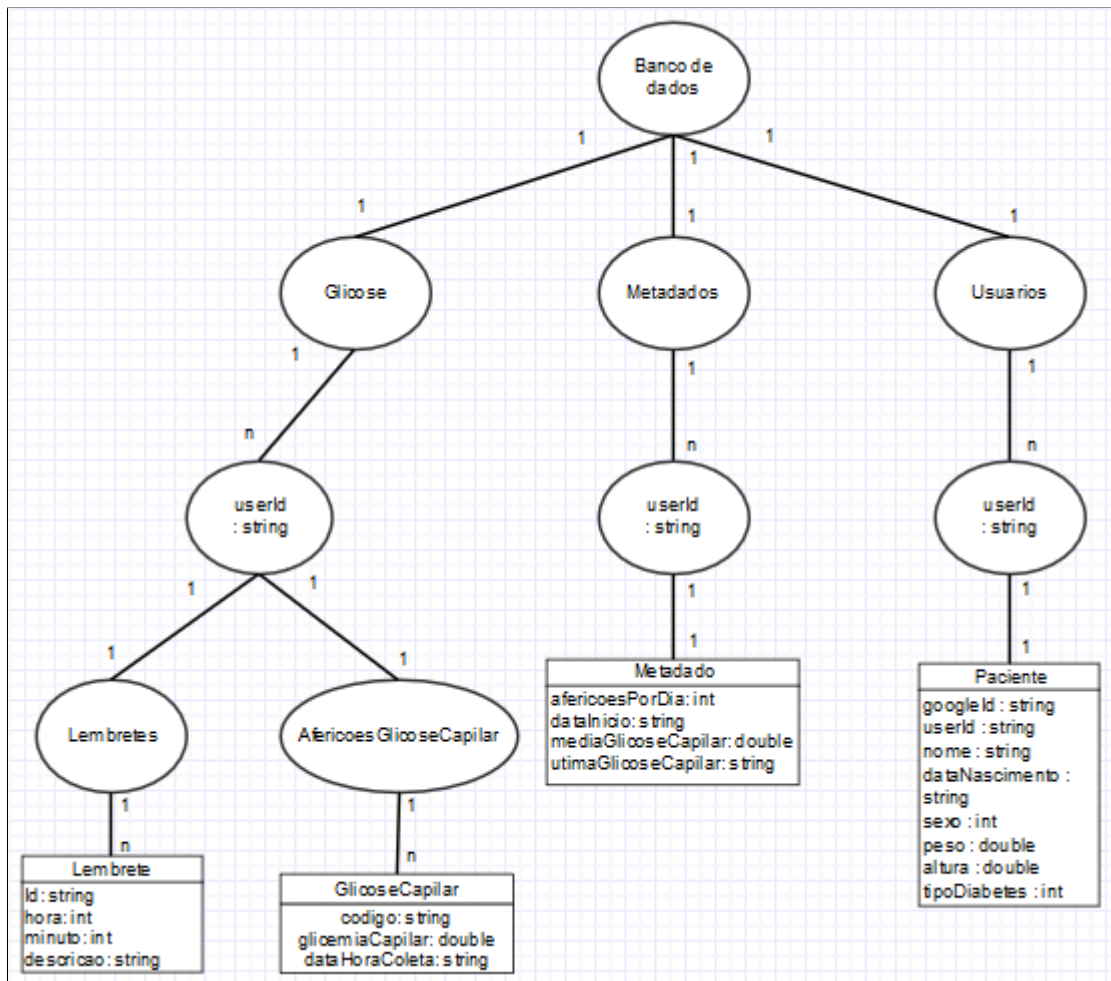


Fonte: Autor.

O banco de dados utilizado neste projeto foi o *Firebase*. Ele é um banco de dados *NoSql*, e que portanto, os seus dados não se encaixam nos modelos de Entidade-Relacionamento. O *Firebase* representa os seus dados em forma de árvore. Isto acontece por que os seus dados são armazenados em objetos *JSON*, que é uma estrutura de dados hierárquica, muito usada na internet (FIREBASE, 2018). Portanto, os dados foram representados em uma estrutura de dados

em árvore, apresentado na Figura 6.10.

Figura 6.10 – Modelo de dados do aplicativo TiaBete.

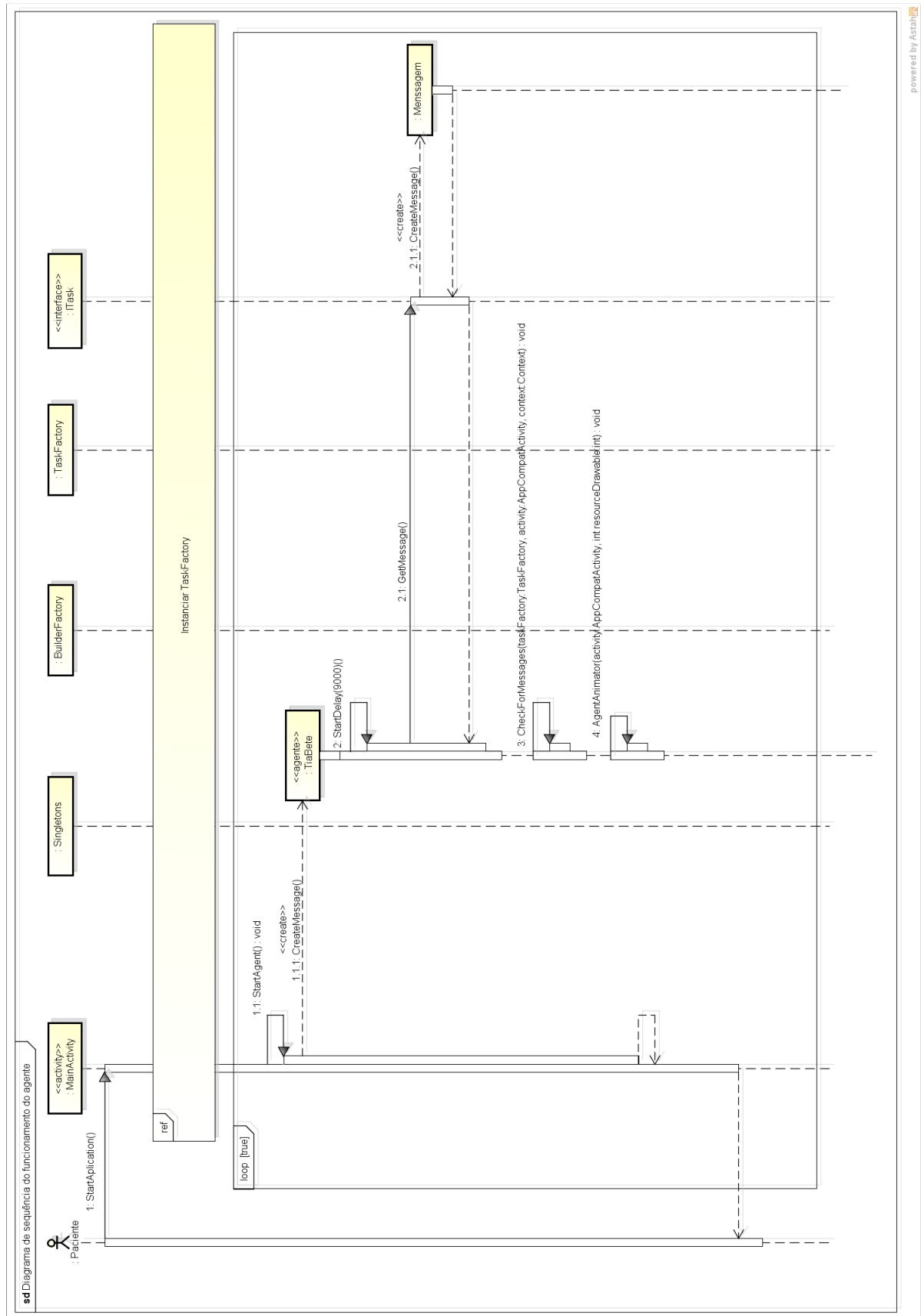


Fonte: Autor.

Para entender o comportamento dos sistemas em tempo de execução, não bastam apenas diagramas de código estático, mas sim de modelos que representem a comunicação dinâmica entre os objetos durante a sua execução (PRESSMAN; MAXIM, 2016).

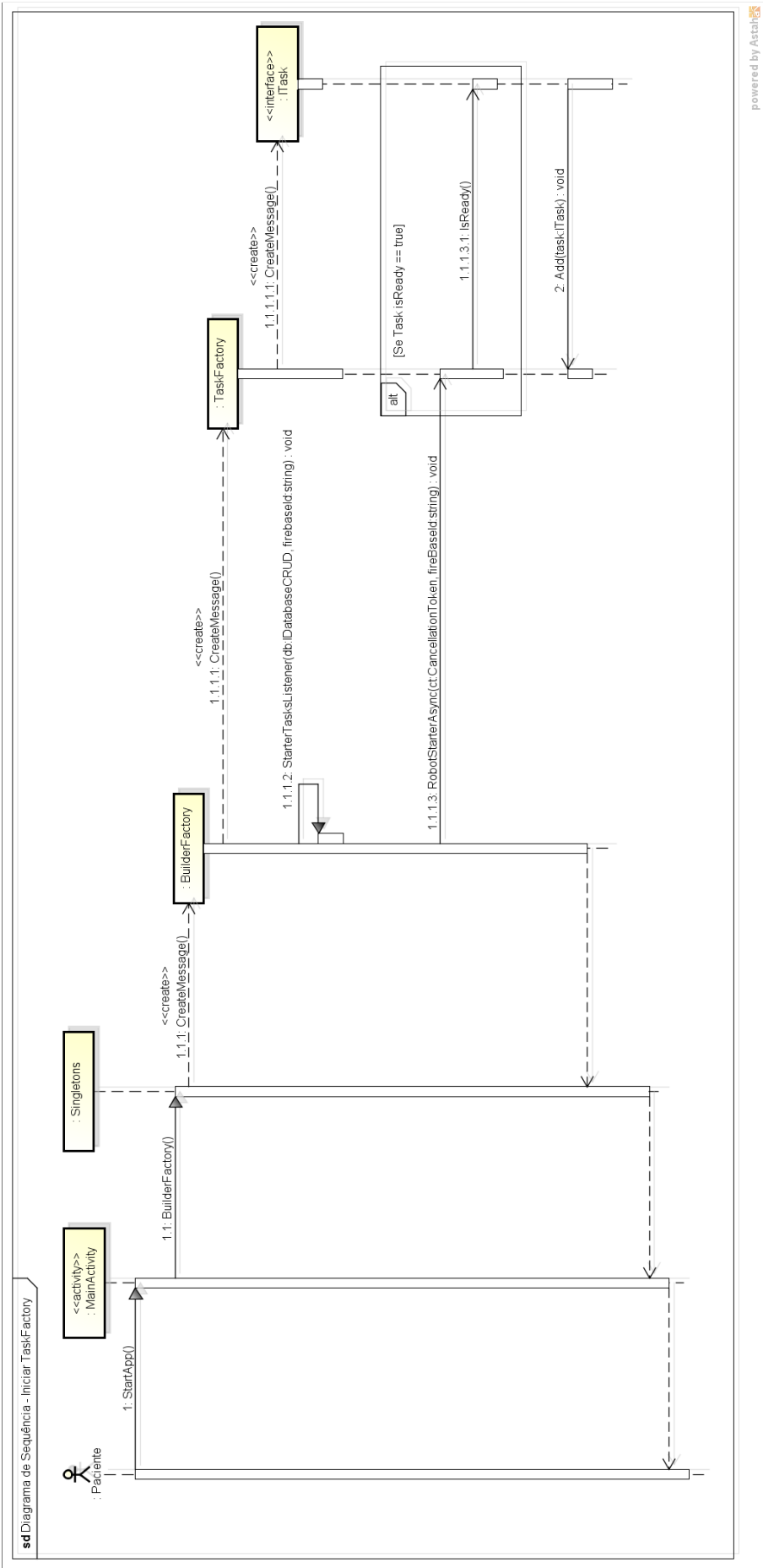
A UML dispõe de modelos como o Diagrama de Sequência, que demonstra a execução dos objetos em relação ao tempo, e o Diagrama de Máquina de Estados, que foca no estado de um componente ou objeto do sistema em relação aos seus atributos em determinado momento da execução. A Figura 6.11 mostra o processo de funcionamento do agente TiaBete. Ele foi dividido em dois diagramas de sequência, para melhor entendimento.

Figura 6.11 – Diagrama de Sequência do Agente Tiabete.



Fonte: Autor.

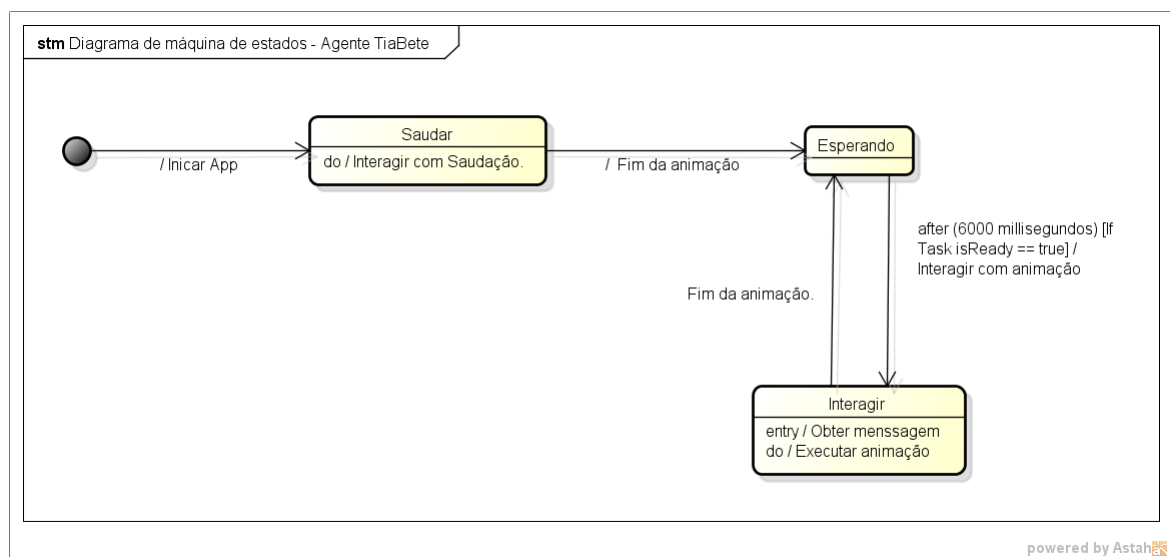
Figura 6.12 – Diagrama de Sequência Instanciar *TaskFactory*.



Na Figura 6.11 mostra o comportamento dos objetos durante a execução. O compar-timento nomeado de "Instanciar *TaskFactory*" acopla um comportamento externo que é feito antes do agente iniciar a sua execução. Nele as tarefas são carregadas de forma assíncrona e disponibilizadas para o agente. A Figura 6.12, exibe o Diagrama de Sequência Referenciado como "Instanciar *TaskFactory*".

O agente TiaBete interage com o usuário por meio de duas animações que exprimem a emoção de felicidade e tristeza. Para isto o agente possui três estados, estes estados são apresentados na Figura 6.13.

Figura 6.13 – Diagrama de Máquina de Estados do Agente TiaBete.



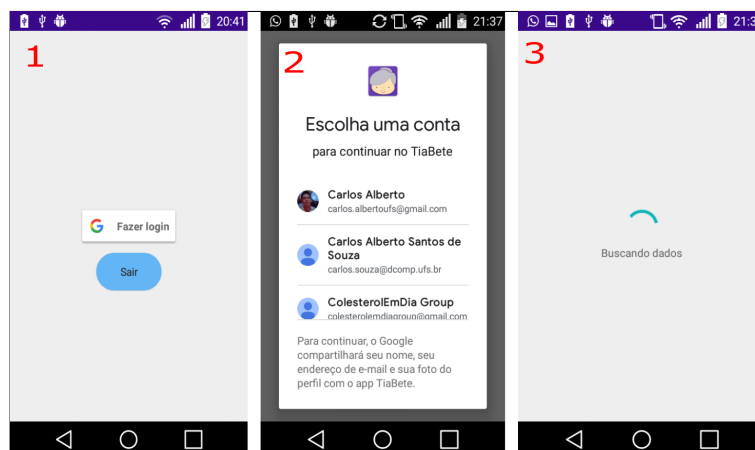
Fonte: Autor.

Por fim, a 1ª iteração foi construída com base nos artefatos reunidos nas fases de Concepção e Elaboração. A aplicação TiaBete foi então disponibilizada para a fase de testes internos do *Google Play Console*.

6.4.3 Demonstração

Ao fim da 1ª Iteração do *RUP*, foi gerado um protótipo executável da TiaBete. Nesta seção é apresentado o passo a passo do fluxo implementado das funcionalidades de Login, Lembretes, Cadastro da Glicemia capilar, Exibição de dicas e Interação com o usuário por meio da avaliação da Hemoglobina Glicada do paciente. A Figura 6.14 mostra o passo a passo do Login da aplicação. Nela cada tela está numerada segundo a seguinte ordem de execução: no passo 1 o usuário clica no botão fazer login; no passo 2 ele então escolhe a conta ao qual vinculará os seus dados; e no passo 3 o sistema faz a busca pelos seus dados.

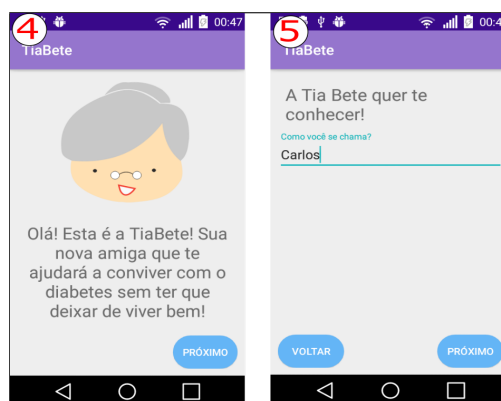
Figura 6.14 – Funcionalidade Login da TiaBete.



Fonte: Autor.

Se o usuário não possuir um perfil cadastrado ao TiaBete, ele será redirecionado para o caso de uso de cadastro de perfil de usuário. Nele o usuário informa o seu nome, data de nascimento, peso, altura, sexo e tipo do diabetes. A Figura 6.15 demonstra as telas exibidas durante o cadastro. No passo 4, é exibida uma tela de preparação do usuário, para então começar o cadastro, onde é inserido um valor para cada tela como mostra no passo 5.

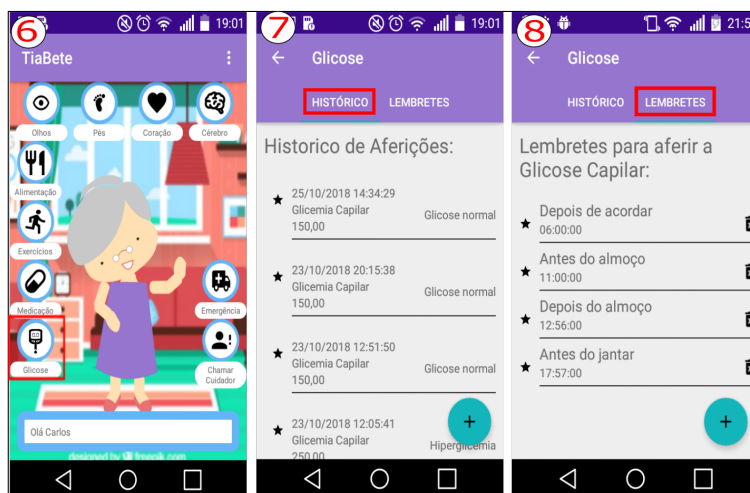
Figura 6.15 – Funcionalidade Cadastro de Paciente.



Fonte: Autor.

Caso o usuário já tenha um perfil cadastrado ou quando finalizar o cadastro do seu perfil ele é então levado a tela principal representado pela Figura 6.16. Neste momento o agente Tiabete entra no estado de "saudação", onde ela saúda o usuário com uma animação de saudação, depois ela entra no seu estado de "esperando" conforme demonstrado no diagrama 6.13.

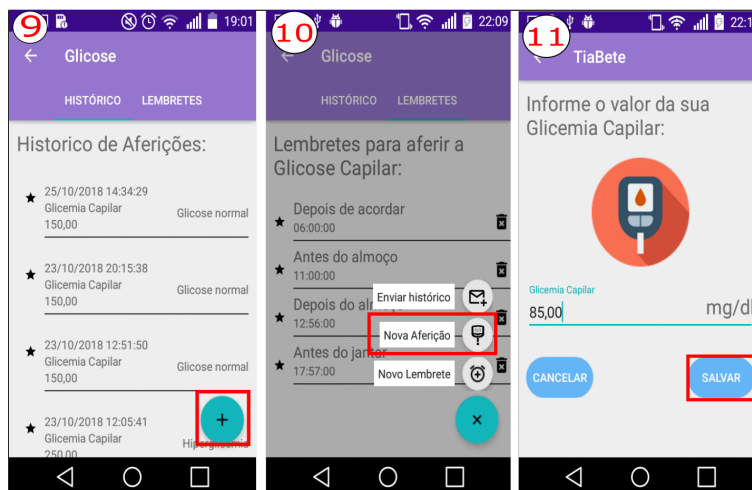
Figura 6.16 – Módulo Glicose.



Fonte: Autor.

Como estamos na primeira iteração, o único módulo construído é o de Glicose. Como mostra a Figura 6.16, ao clicar no ícone Glicose no passo 6, a tela do passo 7 é então exibida, com destaque para o histórico de Aferições da glicemia Capilar e no passo 8, destaque para os Lembretes. Dando continuidade, para poder fazer o cadastro de uma Aferição de Glicemia Capilar, é feito o passo a passo descrito na Figura 6.17.

Figura 6.17 – Cadastro de Aferição da Glicemia Capilar.

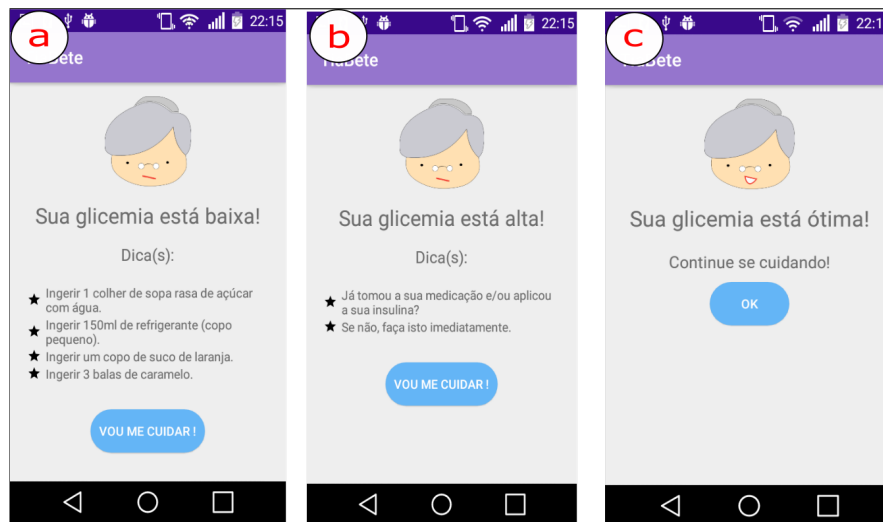


Fonte: Autor.

O usuário clica no botão com sinal de "+" como mostra o passo 9 da Figura 6.17. Uma lista de opções é exibida para o usuário onde ele deve escolher a opção nova aferição como mostra o passo 10. Então é exibida a tela do passo 11, onde o paciente informará o valor da Glicemia Capilar que ele obteve em seu teste glicêmico. Logo depois ele deve clicar em salvar para finalizar o processo.

Após ter salvo o registro da sua Glicemia Capilar, o sistema exibe uma tela que dá ao usuário um *feedback* do seu nível de glicose no sangue. Esse *feedback* pode ser feito em forma de dicas ou uma frase de encorajamento. Ele é apresentado na Figura 6.18.

Figura 6.18 – Dicas dadas pelo aplicativo.

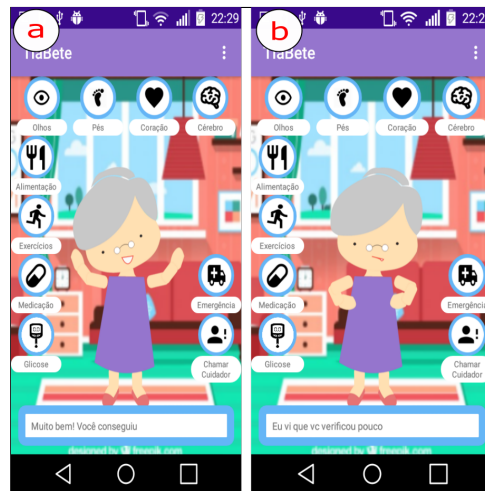


Fonte: Autor.

A Figura 6.18 exibe as três telas diferentes que podem aparecer a depender do nível da glicose do paciente. Se o valor da Glicemia Capilar for menor ou igual a 70 ml/dl, então a Figura 6.18a é exibida. Caso o valor da Glicemia Capilar seja maior ou igual a 180 ml/dl, a Figura 6.18b é mostrada ao paciente. Estes valores são recomendados pela SOCIEDADE BRASILEIRA DE DIABETES (2018). Caso o valor caia no intervalo entre 70 ml/dl e 180 ml/dl, então a Figura 6.18c é apresentada.

O agente TiaBete age sobre os dados da Glicemia Capilar que foram salvos. Ele executa *Tasks* ou tarefas do agente, padrão presente em (MARCO; PACE, 2013), que representa o objeto que armazena o conhecimento para que esta tarefa seja executada com êxito. Somente a tarefa de agente "EstimacaoHemoglobinaGlicada" está disponível nesta iteração. É ela que gera as interações que são executadas pelo agente de forma assíncrona e independente do aplicativo. A Figura 6.19 exibe os dois tipos de animações que são executadas pelo agente, que são: feliz (Figura 6.19a) e triste (Figura 6.19b).

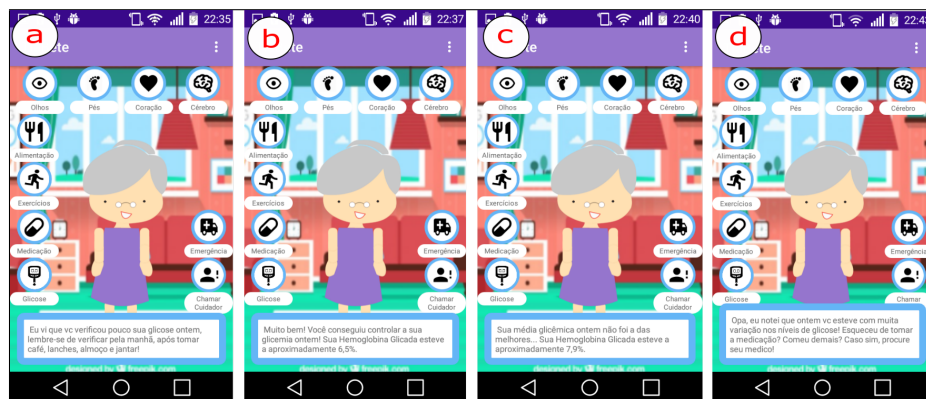
Figura 6.19 – Animações TiaBete.



Fonte: Autor.

E acompanhado as animações descritas na Figura 6.19, as tarefas de agentes mapeiam mensagens às animações da TiaBete. Cada tarefa possui o seu conjunto de mensagens e elas decidem quando cada uma deve ser lançada. A Figura 6.20 exibe as possíveis mensagens lançadas pela tarefa de agente EstimacaoHemoglobinaGlicada.

Figura 6.20 – Mensagens da tarefa EstimacaoHemoglobinaGlicada.



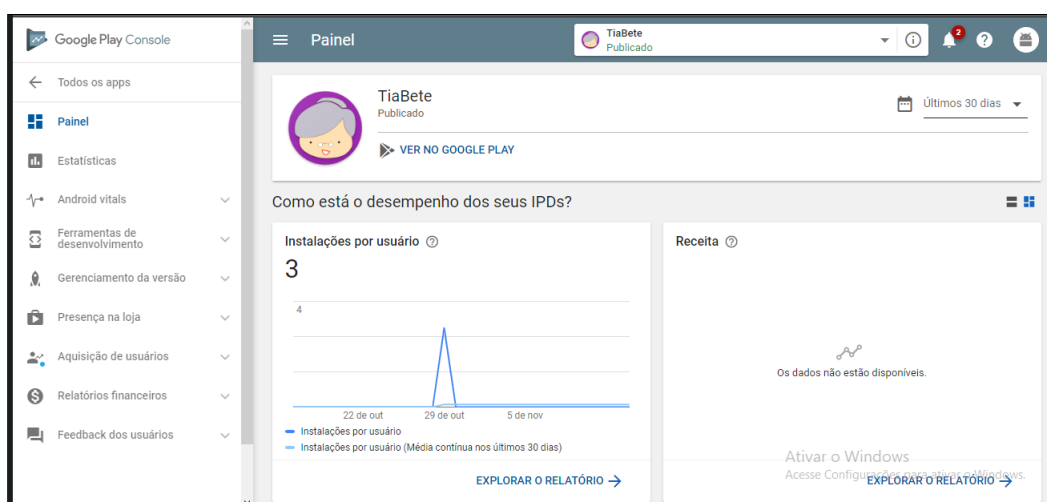
Fonte: Autor.

As mensagens apresentadas na Figura 6.20 são lançadas ao usuário de forma autônoma, sem a necessidade de alguma ação do usuário que a dispare. Elas mudam a depender do perfil do usuário, neste caso, um perfil de Glicose Capilar. A estimativa é feita com base no cálculo sugerido pela (ANERICAM DIABETES ASSOCIATION, 2018), onde a Hemoglobina Glicada é estimada através da seguinte regressão linear simples: $(eAG + 46.7)/28.7 = A1C$, onde *eAG* significa *estimated Average Glucose*, que representa a média estimada da Glicose Capilar enquanto que *A1C* é o valor da Hemoglobina Glicada. Desta forma, é possível traçar uma linha de referência no tratamento do Diabetes.

6.5 Transição

Ao Final da Iteração uma versão *Release* foi criada e enviada para o *Google Play Console*³, que é o gerenciador de versões de aplicativos da *Google Play*. Para ter acesso a ele é preciso ter uma licença de desenvolvedor da *Google*. A Figura 6.21 mostra a os detalhes do aplicativo dentro do Google Play Console.

Figura 6.21 – Visão do Aplicativo TiaBete dentro do Google Play Console.



Fonte: Autor.

Na Figura 6.21 no canto esquerdo encontra-se a barra de ações do *Google Console*, nela encontra-se a aba de gerenciamento de versão. É nesta aba que são criados os tipos de versões que serão publicadas na loja, são eles: testes Internos, testes Alfa fechado, testes Beta Aberto e por fim Produção. Como esta aplicação foi projetada com o objetivo de ser um protótipo executável. Ela foi publicada como versão de Teste Interno.

6.6 Considerações do capítulo

Neste capítulo foi detalhado o processo de desenvolvimento do aplicativo TiaBete. Nele foi possível mostrar o uso do processo de desenvolvimento de software *RUP*, utilizado como base, as fases do Processo Unificado para acoplar as atividades de construção de um Sistema Centrado em Agente para dispositivos móveis. Após a análise inicial do projeto e produto, observou-se a necessidade de dividi-lo em nove iterações, ao qual, a iteração de número 1, foi escolhida para ser iniciada primeiro. A 1ª iteração foi segmentada dentro das quatro fases do *RUP*, onde foram gerados artefatos e código executável. Na fase de Concepção foi gerado um Diagrama de *Use Case* que delimitou as funcionalidades do protótipo para a 1ª iteração. Ao finalizar a etapa de

³ <https://play.google.com/apps/publish/?hl=pt-BR>

Concepção, deu-se início a fase de Elaboração, onde foi gerado a arquitetura de software, que se baseou em uma arquitetura em camadas. Na fase de Construção, foram gerados códigos fonte que obedeceram os requisitos dos artefatos gerados nas fases anteriores. Nela foram criados a aplicação e a assistente virtual TiaBete, que por sua vez fora construída sobre os conceitos de um agente de software. Depois de gerados os seus Diagramas de Classes de projeto, modelo de dados, Diagramas de Sequência e Diagramas de Máquinas de Estados, a iteração foi concluída. Na etapa de Transição, o protótipo resultante foi submetido ao gerenciador de aplicativos da *Google* e está em pleno teste interno. No próximo capítulo são dadas as considerações do trabalho. Nele é reiterado o propósito deste trabalho, o problema a que ele buscou resolver, o processo utilizado para tentar resolver o problema, os resultados obtidos e os trabalhos futuros que podem ser desenvolvidos a partir deste estudo.

7 Considerações Finais

O propósito deste trabalho foi construir um protótipo de uma Assistente Virtual para o autocuidado de pacientes com Diabetes, implementado como um aplicativo móvel com base em um Agente de Software. Este objetivo foi alcançado por meio do uso de conceitos e dados reunidos e embasados na fundamentação teórica, elaboração de revisão sistemática, estudo de aplicações semelhantes, pesquisas de campo e por fim, graças a execução das tarefas de desenvolvimento do aplicativo.

O Diabetes *Mellitus* é uma doença crônica que pode levar uma pessoa à morte caso não seja diagnosticada e tratada a tempo. Os três tipos de diabetes mais comuns são o tipo 1, tipo 2 e o diabetes gestacional, sendo que a maioria dos diabéticos são do tipo 2. A falta de controle da glicemia com o tempo leva a complicações da diabetes e assim criar comorbidades e sequelas.

O autocuidado é a melhor forma de se ter uma vida saudável e com qualidade. Existem diversas tecnologias que facilitam o alto cuidado de pessoas com diabetes, e dentre elas está o uso de aplicativos móveis para o gerenciamento da glicemia. Para o desenvolvimento de um aplicativo móvel foi necessário conhecer bem o paradigma de desenvolvimento multi compilado, pois, foi este o utilizado em sua construção. Foi pesquisado sobre a arquitetura da Plataforma *Android* e suas camadas.

Para poder criar uma aplicação móvel de forma multi compilada foi necessário a utilização da plataforma de desenvolvimento *Xamarin*, que é desenvolvida em C e também é acoplada ao *Visual Studio*. Para a construção de uma Assistente Virtual, foram pesquisados artigos e conceitos sobre Agentes de Software e os seus tipos, pois, foi notada a sua utilidade na solução do problema. Foram feitas buscas sobre conceitos que auxiliassem no processo de desenvolvimento de software, onde foi visto sobre o modelo de Processo Unificado da *Rational* o RUP. Nele foram detalhadas as suas fases no processo que veio a compor o processo de construção de software utilizado. Para tornar possível a construção de um agente dentro do processo RUP, foram incluídas atividades metodológicas sugeridas por diversos especialistas da área de desenvolvimento de agentes de software focados em aplicações móveis. E por fim, para poder auxiliar na elicitação dos requisitos do aplicativo, foi utilizadas técnicas de modelagem de objetivos por meio dos modelos *ROADMAP* e modelos de *i** Objetivos citados em (CHEAH et al., 2017).

A pesquisa de campo foi feita com base nos conceitos da fundamentação teórica e da revisão sistemática. Durante dois meses e meio foram colhidas informações de diversos pacientes com diabetes por meio de formulários pela internet. A pesquisa mostrou que dentro os participantes, a maioria era jovem com idade menor que 20 anos, mostrou também que a maioria tinha diabetes do tipo 1. O sistema operacional *Android* foi o mais utilizado pelos entrevistados.

O planejamento alimentar foi a funcionalidade de aplicação móveis mais importante, segundo a pesquisa. Dentre os problemas mais comuns, a falta de conhecimento sobre as métricas e medidas presentes nos aplicativos foi a mais votada.

Dentro da revisão sistemática pode-se levantar estudos que mostram os trabalhos que outros pesquisadores que serviram de base para o projeto. O objetivo principal da revisão foi de elicitare estudos que descrevem intervenções em tratamentos do Diabetes por meio de aplicações móveis. A seleção de estudos primários foi feita dentro das seguintes bases de dados digitais: *IEEE Xplore*, *PubMed*, *Medline*, *ScienceDirect*, e *Scopus*. Após as etapas de inclusão e exclusão de artigos, sobraram cerca de seis artigos, ao qual foram extraídos dados importantes sobre seus estudos. A extração de dados reuniu as principais técnicas de tratamento que utilizam as aplicações móveis. Estes dados mostram que é possível conciliar o tratamento de complicações da diabetes com aplicativos móveis. Mostrou também que a atenção aos exercícios físicos gerenciamento da alimentação e monitoramento da glicose são as funcionalidades mais comuns aos aplicativos.

No estudo de aplicações semelhantes, foram selecionados três aplicativos da categoria Medicina, gratuito, com nível de avaliação de até 4 estrelas e com quantidade de downloads com Mais de 10.000 downloads. Os aplicativos selecionados foram: mySugr: Diário da Diabetes, Diabetes:M e o SocialDiabetes. Foram descritas as características de cada um deles com base em suas funcionalidades e usabilidade. Ao final da análise, foi gerado um quadro que mostra um comparação entre as suas principais funcionalidades.

A pesquisa de campo foi feita com base nos conceitos da fundamentação teórica e da revisão sistemática. Durante dois meses e meio foram colhidas informações de diversos pacientes com diabetes por meio de formulários pela internet. A pesquisa mostrou que dentro os participantes, a maioria era jovem com idade menor que 20 anos, mostrou também que a maioria tinha diabetes do tipo 1. O sistema operacional *Android* foi o mais utilizado pelos entrevistados. O planejamento alimentar foi a funcionalidade de aplicação móveis mais importante, segundo a pesquisa. Dentre os problemas mais comuns, a falta de conhecimento sobre as métricas e medidas presentes nos aplicativos foi a mais votada.

Após reunir conceitos ligados ao Diabetes, aplicações móveis e agentes de software, trabalhos semelhantes, ter obtido uma pequena amostra do público alvo e uma análise de aplicações semelhantes, tornou possível a construção de um protótipo de aplicativo móvel baseado em um agente de software focado no autocuidado do Diabetes. A etapa de construção do aplicativo foi subdividida dentro das 4 fases do *RUP*, onde foram acopladas atividades metodológicas necessárias para a construção do agente de software. Na fase Concepção, foram elicitados os Requisitos Funcionais, Não- Funcionais da aplicação e seus atores e envolvidos no projeto. A partir destes, foi gerado um Diagrama de *Use Case* que reuniu todas as funcionalidades do aplicativo. Na fase de Elaboração, foi observado a necessidade de segmentar o projeto dentro de iterações para que fosse possível entregar um protótipo executável do produto dentro do prazo,

o projeto foi dividido em 9 iterações. Desta forma, foi gerado um Diagrama de Use Case da 1ª iteração contemplando um conjunto menor de funcionalidades. Após esta etapa, foi produzido o Diagrama de Classes de domínio que tem o objetivo de dar uma visão conceitual do sistema. Para manter compatibilidade com outras plataformas foi desenvolvida uma arquitetura de software em camadas. Na etapa de Construção, foram implementadas as funcionalidades descritas no Diagrama de Caso de Uso da 1ª iteração. Nesta etapa foram produzidos os artefatos: Diagrama de Classes de Projeto da aplicação, Diagrama de Classes de projeto do Agente TiaBete, um modelo de dados da aplicação, dois Diagramas de Sequência do Agente TiaBete e um Diagrama de Máquina de Estados do Agente TiaBete. Ao final da etapa de construção foi feita a demonstração do protótipo do aplicativo. Na etapa de Transição foi mostrada a publicação do aplicativo como uma versão de teste interno na *Google Play*.

A construção do protótipo apresentado abre novos caminhos para novas soluções de software para o tratamento de diversos tipos de doenças. Ele traz também um conjunto de experiências aprendidas durante o desenvolvimento de sua arquitetura que facilitara a inclusão de novos componentes e utilização de sensores. Fornece uma nova forma de pensamento quanto a soluções de software para aplicações móveis baseados em Agentes de Software, demonstrando técnicas de elicitação de requisitos e modelagem do domínio de forma que os *stakeholders* e time de desenvolvimento possam compreender o que será produzindo produzindo.

Como trabalhos futuros sugere-se a execução das iterações restantes do aplicativo, gerar atualizações nos modelos, buscar por padrões de projetos para Agentes de Software e pesquisas de métodos para identificação de padrões nos dados. Sugere-se também criação de estudos sobre o gerenciamento da Glicose com base em informações da Glicose Pós-Prandial, Glicose em Jejum e Hemoglobina Glicada, dar continuidade a pesquisa por formas de gerenciamento inteligente do tratamento do Diabetes quanto a Alimentação, Exercícios, Medicação, Olhos, Coração, Rins, Cérebro e Pé Diabético.

Referências

- ALOTAIBI, M. M.; ISTEPANIAN, R.; PHILIP, N. A mobile diabetes management and educational system for type-2 diabetics in Saudi Arabia (SAED). *mHealth*, AME Publishing Company, v. 2, p. 33, aug 2016. ISSN 2306-9740.
- AMERICAN DIABETES ASSOCIATION. ADA - Standards of Medical Care in Diabetes—2015. *Diabetes Care*, v. 38, n. January, p. S1–S2, 2015. ISSN 19355548.
- AMERICAN DIABETES ASSOCIATION. *eAG/A1C Conversion Calculator* | American Diabetes Association. 2018. Disponível em: <https://professional.diabetes.org/diapro/glucose{_}c>. Acesso em: 11 out 2018.
- APPANNIE. *Global App Downloads Surpassed 175 Billion in 2017*. 2018. 163 p. Disponível em: <<https://www.appannie.com/en/insights/market-data/global-app-downloads-2017/>>. Acesso em: 29 mar 2018.
- AVAZU. *Global Internet Industry Research*. 2016. Disponível em: <[http://avazuinc.com/wordpress/wp-content/download/en/Global%20Internet%20Industry%20Research%20-%20Brazil\(Full%20Edition\).pdf](http://avazuinc.com/wordpress/wp-content/download/en/Global%20Internet%20Industry%20Research%20-%20Brazil(Full%20Edition).pdf)>. Acesso em: 28 mar 2018.
- BADHAM, J. et al. Developing agent-based models of complex health behaviour. *Health and Place*, Elsevier Ltd, v. 54, n. August, p. 170–177, 2018. ISSN 18732054.
- BELGHIAT, A. et al. Mobile agent-based software systems modeling approaches: A comparative study. *Journal of Computing and Information Technology*, v. 24, n. 2, p. 149–163, 2016. ISSN 13301136.
- BIOLCHINI, J. et al. Systematic Review in Software Engineering. *Engineering*, v. 679, n. May, p. 1–31, 2005. ISSN 18650929.
- BIØRN-HANSEN, A.; MAJCHRZAK, T. A.; GRØNLI, T.-M. Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. *Proceedings of the 13th International Conference on Web Information Systems and Technologies*, n. Webist, p. 344–351, 2017.
- BRANDÃO, D. S.; PINHEIRO, M. C. (Org.). *Estratégias para o cuidado da pessoa com doença crônica: diabetes mellitus*. 1. ed. Brasília - DF: [s.n.], 2016. v. 35. 709–17; discussion 717 p. ISSN 1530-891X. ISBN 9788533420595.
- CHEAH, W. et al. Agent-oriented requirement engineering for mobile application development. *International Journal of Interactive Mobile Technologies*, v. 11, n. 6, p. 32–48, 2017. ISSN 18657923.
- DESVEAUX, L. et al. A randomized wait-list control trial to evaluate the impact of a mobile application to improve self-management of individuals with type 2 diabetes: a study protocol. *BMC Medical Informatics and Decision Making*, BioMed Central, London, v. 16, p. 144, nov 2016. ISSN 1472-6947.
- DINSOREANU, M.; SALOMIE, I.; PUSZTAI, K. On the design of agent-based systems using UML and extensions. *Proceedings of the International Conference on Information Technology Interfaces, ITI*, p. 205–210, 2002. ISSN 13301012.

FERNANDES, J. M.; FERREIRA, A. L. *Quality attributes for mobile applications*. [S.l.]: IGI Global, 2016. 141–154 p. ISBN 9781466699175; 1466699167; 9781466699168.

FIREBASE. *Estruturar seu banco de dados | Firebase Realtime Database | Firebase*. 2018. Disponível em: <<https://firebase.google.com/docs/database/android/structure-data?authuser=0>>. Acesso em: 11 out 2018.

GAMA, M. P. R. Do milagre canadense do século xx às esperanças de cura do século xxi (editorial). *Endocrinologia & Diabetes Clínica e Experimental*, 2 (2), p. 3–5, 2002.

GOOGLE DEVELOPERS. *Arquitetura da plataforma | Android Developers*. 2018. Disponível em: <<https://developer.android.com/guide/platform/>>. Acesso em: 07 out 2018.

GUEDES, G. T. A.; VICARI, R. M. Applying a UML profile in the requirements modeling to multi-agents systems. *Proceedings - 7th International Conference on the Quality of Information and Communications Technology, QUATIC 2010*, p. 216–221, 2010. ISSN 17519918.

HSU, W. C. et al. Utilization of a Cloud-Based Diabetes Management Program for Insulin Initiation and Titration Enables Collaborative Decision Making Between Healthcare Providers and Patients. *Diabetes Technology & Therapeutics*, Mary Ann Liebert, Inc., 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 18, n. 2, p. 59–67, feb 2016. ISSN 1520-9156.

IBGE. *Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal*. Rio de Janeiro: IBGE, 2016. ISBN 9788524044458.

IBM. Rational Unified Process Best Practices for Software. *Development*, p. 1–21, 2004. ISSN 07407459.

INTERNATIONAL DIABETES FEDERATION. *IDF Diabetes Atlas*. 8th editio. ed. [s.n.], 2017. ISBN 9782930229874. Disponível em: <<https://www.idf.org/e-library/welcome.html>>. Acesso em: 14 abr 2018.

JEZIC, G. et al. *Agents and Multi-Agent Systems: Technologies and Applications 2018: Proceedings of the 12th International Conference on Agents and Multi-Agent Systems: Technologies and Applications (KES-AMSTA-18)*. [S.l.]: Springer, 2018. v. 96.

JIN, L.; LIANG, X. Modeling of instant messaging system based on RUP and UML. *Proceedings - 2016 International Conference on Computational Intelligence and Applications, ICCIA 2016*, p. 61–66, 2016. ISSN 2165-1701.

JOBE, W. Native Apps Vs. Mobile Web Apps. *International Journal of Interactive Mobile Technologies (IJIM)*, v. 7, n. 4, p. 27, 2013. ISSN 1865-7923.

JONGH, D. Cochrane Database of Systematic Reviews Mobile phone messaging for facilitating self-management of long-term illnesses (Review). n. 12, 2012.

KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. TR/SE-0401, p. 28, 2004. ISSN 13537776.

LATIF, M. et al. Cross platform approach for mobile application development: A survey. *2016 International Conference on Information Technology for Organizations Development (IT4OD)*, p. 1–5, 2016.

- LATIF, M. et al. Review of mobile cross platform and research orientations. *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems, WITS 2017*, p. 0–3, 2017.
- LECHETA, R. R. *Google Android-3ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK*. [S.l.]: Novatec Editora, 2013.
- LEITÃO, P.; KARNOUSKOS, S. *Industrial Agents: Emerging Applications of Software Agents in Industry*. [S.l.]: Morgan Kaufmann, 2015.
- LISANGELA, R. O Diabetes Mellitus E a Hemoglobina Glicada E Sua Correlação Com a Glicemia Diabetes Mellitus and the Glycated Hemoglobin and Its Correlation. *Revista Saúde e Desenvolvimento*, vol. 3, p. 18, 2013.
- MAGLALANG, D. D.; YOO, G. J. A cceptability and C ultural R elevance of a M obile H ealth L ifestyle I ntervention for F ilipinos with T ype 2 D iabetes. v. 27, n. 2, p. 143–155, 2017.
- MARCO, A. D.; PACE, S. Model-driven approach to Agilla Agent generation. *2013 9th International Wireless Communications and Mobile Computing Conference, IWCMC 2013*, p. 1482–1487, 2013. ISSN 2376-6492.
- MICROSOFT. *Introdução ao desenvolvimento móvel - Xamarin | Microsoft Docs*. 2014. Disponível em: <<https://docs.microsoft.com/pt-br/xamarin/cross-platform/get-started/introduction-to-mobile-development>>. Acesso em: 07 out 2018.
- MICROSOFT. *Desenvolvimento de Aplicativos Xamarin com Visual Studio | Visual Studio*. 2018. Disponível em: <<https://visualstudio.microsoft.com/pt-br/xamarin/>>. Acesso em: 07 out 2018.
- MONOPROJECT. *About Mono | Mono*. 2018. Disponível em: <<https://www.mono-project.com/docs/about-mono/>>. Acesso em: 07 out 2018.
- MYSUGR GMBH. *mySugr: Diário da Diabetes*. Google Play, 2018. Disponível em: <<https://play.google.com/store/apps/details?id=com.mysugr.android.companion>>. Acesso em: 11 out 2018.
- NETTO, A. P. *Educação em diabetes: um dever de todos*. 2015. Disponível em: <<http://www.diabetes.org.br/publico/artigos-sobre-diabetes/1155-educacao-em-diabetes-um-dever-de-todos>>. Acesso em: 11 abr 2018.
- NUNDY, S. et al. Developing a behavioral model for mobile phone-based diabetes interventions. *Patient Education and Counseling*, Elsevier Ireland Ltd, n. 1, p. 125–132, 2013. ISSN 07383991.
- OHA. *Alliance Overview, Open Handset Alliance*. 2018. Disponível em: <http://www.openhandsetalliance.com/oha/{_}overview.h>. Acesso em: 27 mar 2018.
- OLIVEIRA, J. E. P. de (Org.). *Diretrizes da Sociedade Brasileira de Diabetes 2017-2018*. São Paulo: Clannad, 2017. 383 p. ISBN 9788593746024.
- OOCITIES.ORG. *Sistemas Multiagentes*. 2018. Disponível em: <<http://www.oocities.org/igoryepes/agentes.htm>>. Acesso em: 20 out 2018.
- POPPE, L. et al. Users' thoughts and opinions about a self-regulation-based eHealth intervention targeting physical activity and the intake of fruit and vegetables: A qualitative study. *PLoS ONE*, Public Library of Science, San Francisco, CA USA, v. 12, n. 12, p. e0190020, dec 2017. ISSN 1932-6203. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5739439/>>.

- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016.
- QUE, P.; GUO, X.; ZHU, M. A Comprehensive Comparison between Hybrid and Native App Paradigms. *Proceedings - 2016 8th International Conference on Computational Intelligence and Communication Networks, CICN 2016*, p. 611–614, 2017.
- RUSSELL, S.; DOYLE, O.; COLLIER, R. W. Developing Android applications using agent-oriented programming. *Proceedings of the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2017*, v. 2018-January, p. 1–7, 2018.
- RUSSELL, S. J. et al. *Artificial Intellegence - A Modern Approach*. 3 ed. ed. [S.l.]: Pearson Prentice Hall, 2014. 1080 p. ISSN 09252312. ISBN 0131038052.
- SIMÕES, D. D.; PEREIRA, J. C. *Sistemas Operacionais Móveis - Android X IOs*. 2014.
- SIRMA MEDICAL SUSTEMS. *Diabetes:M*. Google Play, 2018. Disponível em: <<https://play.google.com/store/apps/details?id=com.mydiabetes>>. Acesso em: 11 out 2018.
- SOCIAL DIABETES. *SocialDiabetes*. Google Play, 2018. Disponível em: <<https://play.google.com/store/apps/details?id=com.socialdiabetes.android>>. Acesso em: 11 out 2018.
- SOCIEDADE BRASILEIRA DE DIABETES. *Sociedade Brasileira de Diabetes - Diagnóstico e Tratamento*. 2018. Disponível em: <<http://www.diabetes.org.br/publico/diabetes/diagnostico-e-tratamento>>. Acesso em: 14 abr 2018.
- SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 544 p. ISBN 9788579361081.
- STATCOUNTER. *Mobile & Tablet Operating System Market Share Worldwide | StatCounter Global Stats*. 2018. Disponível em: <<http://gs.statcounter.com/os-market-share/mobile-tablet/worldwide/{\#}monthly-201702-201>>. Acesso em: 21 mar 2018.
- STUCKEY, M. I. et al. A lifestyle intervention supported by mobile health technologies to improve the cardiometabolic risk profile of individuals at risk for cardiovascular disease and type 2 diabetes: study rationale and protocol. *BMC Public Health*, BioMed Central, v. 13, p. 1051, nov 2013. ISSN 1471-2458. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3922899/>>.
- SUN, Y.; LI, Z.; XIE, J. Recommendation agents in intelligent query answering systems. *Proceedings of the 8th International Conference on Computer Science and Education, ICCSE 2013*, n. Iccse, p. 441–445, 2013.
- WHO. *Global Report on Diabetes*. 2016. Disponível em: <<http://www.diabetes.org.br/publico/diabetes/diagnostico-e-tratamento>>. Acesso em: 14 abr 2018.

Apêndices

Apêndice A - Questionário usado na pesquisa de campo para elicitación de requisitos do aplicativo.

Este questionário foi utilizado na pesquisa exploratória para identificar as necessidades e os problemas encontrados pelos pacientes durante a utilização de aplicativos para o autocuidado do diabetes. A pesquisa iniciou em 14/02/2018 e finalizou a coleta de dados em 20/04/2018. Foram reunidos dados de 24 pessoas com Diabetes *Mellitus* tipo 1, tipo 2 e diabetes gestacional.

Tia Bete - uma assistente virtual mobile para o autocuidado na Diabetes

Esta pesquisa é voltada para pacientes de qualquer faixa etária com Diabetes do Tipo 1, Tipo 2, Diabetes Gestacional ou Pré-Diabetes.

O email disponibilizado será usado para fins de identificação dos dados e não da pessoa pesquisada. Qualquer pessoa pode fazer a coleta de dados de terceiros, contanto que não utilizem o mesmo email para identificá-lo. Não serão enviadas mensagens automáticas (spam).

O Tia Bete é um aplicativo mobile para Android com foco no tratamento de pacientes com Diabetes. Ele será usado como parte do Trabalho de Conclusão de Curso do aluno Carlos Alberto Santos de Souza, graduando em Sistemas de Informação pela Universidade Federal de Sergipe, sob orientação da Dra. Adicinéia A. de Oliveira e com co-autoria do Dr. Lysandro Pinto Borges. Toda a equipe de pesquisa e desenvolvimento agradece pela sua colaboração.

Ao informar seu e-mail neste questionário, você estará concordando em participar desta pesquisa de campo, onde as informações aqui coletadas serão utilizadas em publicações, apresentações de trabalhos acadêmicos e como base para projetos de pesquisa futuras.

***Obrigatório**

1. Estado: *

Questionário

2. Qual a sua idade? *

Marcar apenas uma oval.

- ☐ Menor de 20 anos
- ☐ Entre 20 e 29 anos
- ☐ Entre 30 e 39 anos
- ☐ Entre 40 e 49 anos
- ☐ Entre 50 e 59 anos
- ☐ Acima de 59 anos

3. Qual é o seu sexo? *

Marcar apenas uma oval.

- ☐ Feminino
- ☐ Masculino

4. Qual o tipo do seu Diabetes? *

Marcar apenas uma oval.

- ☐ Tipo 1
- ☐ Tipo 2
- ☐ Diabetes Gestacional
- ☐ Pré-diabetes

5. Qual a sua renda familiar? **Marcar apenas uma oval.*

- ☐ Menos que um salário-mínimo
- ☐ De um salário-mínimo a R\$1.900,00
- ☐ De R\$2.000,00 a R\$2.900,00
- ☐ De R\$3.000,00 a R\$3.900,00
- ☐ De R\$4.000,00 a R\$4.900,00
- ☐ Acima de R\$5.000,00

6. Qual o nível de importância que você dá ao acompanhamento da Diabetes (controle da glicose, alimentação, prática de exercícios, consultas médicas, etc)? **Marcar apenas uma oval.*

- ☐ Muito importante
- ☐ Importante
- ☐ Mediano
- ☐ Pouco importante
- ☐ Nenhuma importância

7. Você tem acesso a algum smartphone? Qual o Sistema Operacional dele ? **Se possuir mais de um, considere a resposta sobre o smartphone mais utilizado por você.**Marcar apenas uma oval.*

- ☐ Android
- ☐ IOS
- ☐ Windows Phone
- ☐ Outros
- ☐ Não possuo smartphone

8. Com que frequência você utiliza o smartphone? **Marcar apenas uma oval.*

- ☐ Várias vezes ao dia
- ☐ Ao menos uma vez por dia
- ☐ Algumas vezes por semana
- ☐ Algumas vezes no mês
- ☐ Nunca uso

9. Você utiliza algum aplicativo no celular que lhe ajuda no tratamento da Diabetes? **Marcar apenas uma oval.*

- ☐ Sim
- ☐ Não

10. Se sim, qual?

11. Para você, existe alguma dificuldade em utilizar aplicativos para auxiliar no cuidado da Diabetes? **Marque todas que se aplicam.*

- ☐ Não sinto dificuldade
- ☐ Interface não - intuitiva
- ☐ Gráficos difíceis de entender
- ☐ Notificações que incomodam
- ☐ Leitura confusa
- ☐ Falta de conhecimento sobre certas medidas ou métricas
- ☐ Desmotivação para concluir as tarefas impostas pelo aplicativo
- ☐ Exibe informações desnecessárias
- ☐ Outro: _____

12. Qual funcionalidade você acha ser a mais importante para um aplicativo de cuidado da Diabetes? **Marcar apenas uma oval.*

- ☐ Controle de glicemia (hemoglobina glicada, glicemia normal em jejum)
- ☐ Controle de doses de insulina
- ☐ Planejamento alimentar
- ☐ Gerenciamento de exercícios
- ☐ Funções de emergência em caso de Hipoglicemia ou Hiperglicemia
- ☐ Acompanhamento virtual por um médico
- ☐ Acompanhamento dos fatores de risco (peso, feridas não cicatrizadas, "pé diabético", alterações visuais)
- ☐ Outro: _____

13. Se existisse um aplicativo para smartphone que gerenciase todas as suas informações da Diabetes e previsse o risco de comorbidades (pé diabético, problemas de visão, problemas renais, etc), você utilizaria este aplicativo? **Marcar apenas uma oval.*

- ☐ Sim
- ☐ Não

14. Se respondeu SIM à questão anterior. Quanto você estaria disposto a pagar por ele?*Marcar apenas uma oval.*

- ☐ Acima de R\$16,00
- ☐ Entre R\$11,00 e R\$16,00
- ☐ Entre R\$9,00 e R\$ 10,00
- ☐ Menos que R\$9,00
- ☐ Não compraria